


For Reference

NOT TO BE TAKEN FROM THIS ROOM

Ex LIBRIS
UNIVERSITATIS
ALBERTAE NSIS





Digitized by the Internet Archive
in 2023 with funding from
University of Alberta Library

<https://archive.org/details/Alber1972>

THE UNIVERSITY OF ALBERTA

ON-LINE THESAURUS DESIGN FOR AN INTEGRATED INFORMATION SYSTEM

by



FREDRICK M. ALBER

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

SPRING, 1972

Thesis
1972
2

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled ON-LINE THESAURUS DESIGN FOR AN INTEGRATED INFORMATION SYSTEM submitted by Fredrick M. Alber in partial fulfilment of the requirements for the degree of Master of Science.

ABSTRACT

The rapid increase in the volume of published material has led to research directed towards the improvement and investigation of various controlling mechanisms. Two such controls are indexing and classification. Specifically, thesauri serve as vocabulary control mechanisms in the indexing of information, while classification schemes such as L.C., Dewey Decimal, and U.D.C. give control in classification. A combination of these techniques might be helpful in improving retrieval. So far, such efforts have been of a manual nature. This thesis describes the design of an on-line thesaurus as a central tool for classifying, indexing and searching in a computerized information storage and retrieval system. Thesaurus and classification information may be combined and associated storage and retrieval functions applied through computer programs operative in either on-line or batch modes.

ACKNOWLEDGEMENT

I wish to express my sincere appreciation to my supervisor Mrs. D.M. Heaps for her advice, criticism, and guidance throughout the duration of this research.

Also, appreciation must go to Mr. H.S. Heaps for his assistance and continued interest in the project.

The financial assistance supplied through contracts and grants from the Canada Department of Environment and the National Research Council of Canada is gratefully acknowledged.

TABLE OF CONTENTS

| | Page |
|--|------|
| CHAPTER I - INTRODUCTION | 1 |
| 1.1 Multipurpose and Multidisciplinary Information Systems | 1 |
| 1.2 The Thesaurus Concept -- General Background and Definitions | 4 |
| CHAPTER II - LITERATURE REVIEW | 9 |
| 2.1 Introduction | 9 |
| 2.2 Some Existing Thesauri | 9 |
| 2.3 Thesaurus and Classification Linkage | 13 |
| 2.4 Description of Thesaurus Format and Relationships | 14 |
| 2.5 A Thesaurus and Information Retrieval Effectiveness .. | 17 |
| 2.6 Literature Concerned with Thesaurus Concept | 20 |
| 2.6.1 Introduction | 20 |
| 2.6.2 Literature Dealing with Thesaurus Construction | 22 |
| 2.6.3 Actual Thesauri | 26 |
| 2.6.4 Use of a Thesaurus in an Information System ... | 27 |
| 2.6.5 Value of a Thesaurus as an Aid in Indexing, Retrieval, and Classification | 33 |
| 2.6.6 Thesaurus Reviews | 35 |
| 2.6.7 Conclusions | 37 |
| CHAPTER III - CREATION OF A THESAURUS | 39 |
| 3.1 Introduction | 39 |
| 3.2 Construction by Experts | 39 |

| | Page |
|--|------|
| 3.3 Construction by Machine | 43 |
| 3.3.1 Introduction | 43 |
| 3.3.2 Fully Automatic Methods | 43 |
| 3.3.3 Semi-Automatic Methods | 44 |
| 3.3.4 Hierarchy Formation | 45 |
| 3.3.5 Conclusions | 47 |
| 3.4 Construction by Users in a Man-Machine System | 48 |
| CHAPTER IV - THESAURUS PROGRAM | 55 |
| 4.1 Introduction | 55 |
| 4.2 Theoretical Aspects of File Handling Techniques Used . | 55 |
| 4.2.1 Introduction | 55 |
| 4.2.2 Definitions | 56 |
| 4.2.3 Binary Search | 57 |
| 4.2.4 Chaining | 58 |
| 4.2.5 Tree Allocation | 60 |
| 4.3 Method of Application of File Handling Techniques in THESAURI | 65 |
| 4.4 Description of Module THESAURI | 78 |
| 4.4.1 Introduction | 78 |
| 4.4.2 Thesaurus Entry and Copy | 78 |
| 4.4.3 Command Recognition Routine | 82 |
| 4.4.4 Initialization for Construction of New Thesaurus | 83 |
| 4.4.5 Checking of Space | 84 |

| | Page |
|--|------|
| 4.4.6 Binary Search Routine on Term Information | 85 |
| 4.4.7 Movement of Storage Areas | 86 |
| 4.4.8 Addition of Terms and Relationships to the Thesaurus | 88 |
| 4.4.9 Routine to Print File or Portion Thereof | 91 |
| 4.4.10 Routine to Reposition Storage Area TERMSTOR ... | 93 |
| 4.4.11 Initialization for Call to Routine Which Establishes Relationships | 94 |
| 4.4.12 Addition of Relationship Information to POINTABL | 95 |
| 4.4.13 Routine to Change Spelling of Thesaurus Entries | 97 |
| 4.4.14 Routine to Delete Terms and/or Relationships .. | 98 |
| 4.4.15 Garbage Collection | 100 |
| 4.5 Conclusions | 103 |
| CHAPTER V - THE THESAURUS CENTERED SYSTEM | 104 |
| 5.1 Introduction | 104 |
| 5.2 Thesaurus and Classification Scheme Linkage | 105 |
| 5.3 Overall System Configuration | 109 |
| 5.3.1 Introduction | 109 |
| 5.3.2 Module MONITOR | 111 |
| 5.3.3 Module INDEXING | 111 |
| 5.3.4 Module IOMODULE | 115 |
| 5.3.5 Module SEARCH | 116 |
| 5.3.5.1 Introduction | 116 |

| | Page |
|--|------|
| 5.3.5.2 Internal Workings of the Module SEARCH | 118 |
| 5.4 Command Language | 124 |
| 5.5 Command File | 125 |
| 5.6 Sample Sessions | 126 |
| 5.6.1 Introduction | 126 |
| 5.6.2 MONITOR | 127 |
| 5.6.3 THESAURI | 128 |
| 5.6.4 INDEXING | 131 |
| 5.6.5 SEARCH | 133 |
| 5.7 Conclusions | 136 |
| CHAPTER VI - FUTURE CONSIDERATIONS | 137 |
| 6.1 Introduction | 137 |
| 6.2 Storage Requirements | 137 |
| 6.3 Specification of a Data Base Format | 139 |
| 6.4 Modification of Method for Obtaining Thesaurus Reference Number | 140 |
| 6.5 Modification of Method for Obtaining Record Numbers for Storage of POINTABL Information | 141 |
| 6.6 Automatic Modification of Search Strategy | 142 |
| 6.7 Application of Coding | 143 |
| CHAPTER VII - CONCLUSIONS | 144 |
| BIBLIOGRAPHY | 145 |
| APPENDIX | 152 |

LIST OF TABLES

| | Page |
|---|------|
| Table 4-1: Codes and Meanings | 88 |
| Table 4-2: Print Table Entries | 92 |
| Table A-1: Symbols Used in Command Language | 152 |

LIST OF FIGURES

| | Page |
|---|------|
| Fig. 3-1: Term-Document Matrix | 43 |
| Fig. 3-2: Term-Property Matrix | 44 |
| Fig. 4-1: Algorithm for Binary Search | 58 |
| Fig. 4-2: File and Associated Tree Structure | 61 |
| Fig. 4-3: Computer Representation of a Doubly-Chained Tree .. | 63 |
| Fig. 4-4: Memory Map for Tree of Fig. 4-2 | 63 |
| Fig. 4-5: Format of Records Used in Relationship Accounting . | 70 |
| Fig. 4-6: Diagrammatic Representation of Information Accounting | 74 |
| Fig. 4-7: Example of Information Accounting | 75 |
| Fig. 4-8: Format of Sequential File Containing Thesaurus Information | 81 |
| Fig. 5-1: Diagrammatic Representation of Relationship Between Term Information and Files in System | 108 |
| Fig. 5-2: Overall Configuration for System | 110 |
| Fig. 5-3: Diagrammatic Representation of System Capabilities | 112 |
| Fig. 5-4: Record Formats | 114 |
| Fig. 5-5: System Flowchart for Search Procedure | 119 |
| Fig. 5-6: Relationship Between Tables for Module SEARCH | 123 |

CHAPTER I

INTRODUCTION

1.1 Multipurpose and Multidisciplinary Information Systems

The basic purpose of any information system is to place the user in more efficient and direct contact with the data bases of concern to him, and thus to enable him to make more efficient decisions. Various tools have been developed to improve efficiency at both input and output. The techniques of coordinate indexing, the employment of classification schemes, the development of thesauri, the batching of computer profiles, and on-line query languages are all aids of this kind. In general, however, one type of tool has been used with one type of data base and with one form of computer or library application.

If the user's interests range widely then such single-purpose systems limit his access to information. Some information needs are satisfied only by access to a non-homogeneous data base or to several linked data bases. Some material should be indexed and some classified, some accessed immediately, some available, with some delay, in off-line batch. This diversity of needs and material is very evident in interdisciplinary fields, among these are those that deal with urban dwelling problems, control of pollution, or conservation of natural resources.

At present, these areas are of great significance, especially in the formation of national policy. Therefore, combined information handling techniques which would allow access to many data bases, with the material being identified by various methods, would be of great

value. This is especially true if the processes could be automated. There have, however, been very limited attempts to combine techniques; the English Electric Thesaurofacet [3] scheme is an interesting example but it is little known and is not automated.

The work described in this thesis develops automated methods which provide for the combination of information handling techniques and which give access to and control over multidisciplinary data bases. This thesis project arose in connection with the design of an information system to manage natural resource information.

This system had to allow for the classification, indexing and searching of material from a wide variety of sources and a wide variety of fields. Multiple access was needed, but at the same time there seemed to be a need for some form of central control of the system, or for some central switching mechanism. It was decided to design a system that would allow for the interaction of various classification and indexing techniques and to place an on-line thesaurus in control of the system. This thesaurus would be used as the primary aid in classifying, indexing, and searching the water resource data base.

The data base contains material in bibliographic format of non-standard type. The documents include research project descriptions, research grant applications, monographs, journal articles, abstracts of statutes, entire statutes, and so forth. They are indexed and/or classified by L.C., Dewey, or U.D.C. The computer record includes standard bibliographic elements such as author, title, publication data, and is augmented by keywords, contains accession or

location number, or classification numbers or both. For the purpose of this study it is assumed that both keywords and class numbers are available on every document surrogate. At present no abstracts or continuous text are included.

As stated, this material will be accessed and controlled and new documents indexed and/or classified using an on-line thesaurus as an initial entry point (Fig. 5-3). In total a thesaurus, a data base (document surrogates), and a class structure (schedules) are the principal elements of the system. All may be manipulated by the computer. The thesaurus is the central feature of the system. This thesis is principally concerned with the design and implementation of the thesaurus portion of the system.

It is evident that the on-line manipulation of the thesaurus requires careful consideration. The thesaurus must somehow be linked to the data base of mixed bibliographic format to facilitate both the indexing of new data base entries and the searching of existing data base entries for information. It must also be linked to classification codes to allow for searching of the data base with class numbers as well as keywords. Questions arise about what relationships should be permitted in the thesaurus and about how many entries should be allowed under a specified relationship for a given entry. The central concern governing the computer programming is that the user, through the medium of a suitable query language, must be permitted to create, modify, and display a thesaurus or parts of it and this must be done with a reasonable price in computer core and response time.

As stated this research was carried out as a project concerned

with the design of a water resources information system. M.A. Mercier's thesis [48] developed complementary sections of the system.

For convenience, the main themes of the thesis may be grouped in a general way and these will be discussed in the order indicated:

- (1) the thesaurus concept and the literature concerned with it;
- (2) design and implementation of computer programs to allow for thesaurus manipulation to be efficiently handled in an on-line environment; utilization in batch mode is also allowed;
- (3) design and implementation of computer programs to facilitate and make use of the advantages which may be gained by linking a thesaurus and a classification scheme;
- (4) recommendations and summary.

The computer programs associated with the system are written in 360-Assembler and are operative on the IBM 360/67 at the University of Alberta under the Michigan Terminal System (MTS) operating system.

1.2 The Thesaurus Concept -- General Background and Definitions

"Thesaurus" is a Greek word meaning a "storehouse or treasury" but this definition does not adequately explain its meaning in relation to information science and information systems. B.C. Vickery [82] has given a definition which seems to satisfy both the computer and noncomputer worlds. He believes that the term "thesaurus" can have two meanings: (1) any linear list displaying relations between words; and (2) a tool aiding us to pass from textwords in natural language to keywords or codes in a regularized language.

His first definition will be recognized as valid by any person

who has seen Roget's Thesaurus. It can also be applied to special purpose thesauri where, however, the relations are exactly specified. The idea of a word list with defined relationships between terms is imbedded in his first definition.

The second meaning is not readily understood by the layman. However, the information specialist who uses a thesaurus for indexing and searching understands the definition and perceives it to be correct. The second part of the definition indicates the reason for research into the feasibility of totally automated or computer assisted thesaurus construction; this statement in no way undermines the associated or equal importance of the first meaning.

J.C. Costello Jr. [17] has defined the term "thesaurus" in a manner that reinforces Vickery's definition; moreover, his definition of the term "thesaurus" stresses the aspects that are important in this thesis. He says:

By definition, an information retrieval thesaurus is a display of unit concept terms of an index vocabulary in which terms are alphabetically ordered and in which the relationships of each term to the other terms in the index vocabulary are systematically presented.¹

The user of an information retrieval system, either manual or automated, wants to be reasonably certain that the queries he prepares will retrieve information which is relevant to his interests. A search conducted with user chosen keywords on a document collection or data base may, or may not, retrieve information related to the user's interests. Obviously a method by which the user could be assured, to some

1 Costello, J.C. Jr., Systems for the Intellectual Organization of Information Volume VII Coordinate Indexing, New Brunswick, New Jersey, The Rutgers University Press, 1966, Page 90.

degree, that the retrieved information is relevant to his interests would be an invaluable aid in preparing queries for a retrieval system. One such aid is through vocabulary control using a thesaurus.

A definition that appears in the introduction to the first edition of the Thesaurus of ERIC Descriptors [79] gives a further reason for the use of a thesaurus in a specialized system directed towards efficient information retrieval.

An information retrieval thesaurus is a term-association list structured to enable indexers and subject analysts to describe the subject information of a document to a desired level of specificity at input, and to permit searchers to describe in mutually precise terms the information required at output. A thesaurus therefore serves as an authority list and as a device to bring into coincidence the language of documents and the language of questions.²

This definition, and other statements in the introduction to this thesaurus, stress the importance of a thesaurus as a communication tool. It is realized that the thesaurus serves as an aid in attaining agreement between searchers and indexers.

It is evident that a controlled vocabulary, such as a thesaurus, must be used with many types of data bases particularly when one or more of the following conditions exists: (1) the data base covers a wide range of subject matter; (2) the potential users have different backgrounds and information requirements; (3) the data base shows a lack of continuity in term usage, term meanings, and physical appearance of terms (noun forms, plurals, punctuation); and (4) the user is unable to determine all possible avenues of searching for

2 Thesaurus of ERIC Descriptors, Washington, D.C., U.S. Government Printing Office, 1968, Page vii, Introduction.

information. This often occurs if the indexer and searcher are not the same person.

In discussing an information storage and retrieval system Costello has also given reasons for vocabulary control. These reasons are closely related to the above mentioned conditions when they apply to an information retrieval system. Costello's reasons for vocabulary control are: (1) to improve the quality of description of document content at the time of input; (2) to improve the quality of description of desired content at the time of output; and (3) to improve the relevance and recall ratio characteristics of the system.

Thus in information storage and retrieval systems that operate with keyworded information, a thesaurus should be employed in the indexing of documents and other information. The same thesaurus should also be used in preparing queries to the system. The queries then will consist of, for the most part, the same keywords that are used in indexing the documents or information pertinent to the user's interests. K. Sparck Jones [68] makes the following statement:

... it is generally true that the stronger the match between a request and a document, the more chance the document has of being relevant to the request ...³

Thus the use of a thesaurus in both indexing and query formulation provides a congruence between indexing and searching languages. This agreement, in most cases, improves retrieval effectiveness.

The increasing use of time-shared computer facilities and the

3 Sparck Jones, K., "Automatic Thesaurus Construction and the Relation of a Thesaurus to Indexing Terms", ASLIB Proceedings, Volume 22, Number 5 (May 1970), Page 228.

consequent increase in on-line information retrieval systems covering a variety of disciplines makes some form of vocabulary control mandatory. In such information retrieval systems the basic operations performed are searches conducted on data bases. Quite often the type of search conducted is a weighted term search with "questions" to the system consisting of user specified terms, which hopefully are the same as those used to index the documents or information that the user wishes to obtain. In this type of system it is obvious that a thesaurus may often be absolutely necessary to insure optimum matching of these search questions. Furthermore, if such information systems are to operate in on-line mode there is no reason why a computer program could not be constructed to allow a user to develop and manipulate a thesaurus on-line. It should be noted that any formally structured word authority list may be used in the same manner as a thesaurus.

For the past several years, from 1968, researchers at the University of Alberta have been concerned with on-line thesauri [66, 87]. Initially efforts were devoted to implementing test programs written in PL/1, which made it possible to monitor the reactions of test users and to test primitive query languages. These initial efforts will be described briefly in Chapter 3 since they provided necessary background for much of this research. This thesis, however, is concerned with the efficiency of programs, the amount of storage, the linkage of thesaurus and classification schemes, the integration of the thesaurus and classification schemes, and the integration of the thesaurus into a total system.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

This chapter covers the literature that is directly and indirectly related to the thesaurus concept. Sections two, three, and four give examples of various types of thesauri, provide further definitions, and outline the relationships commonly indicated in thesauri. Section five contains a discussion of system effectiveness. This discussion should make the reader aware that: (1) system effectiveness is important and can be evaluated; (2) system effectiveness can be altered by the use of various aids. Section six is concerned with the literature most directly related to the thesaurus concept. The literature having direct bearing on this concept is categorized, and articles within the various categories are discussed.

2.2 Some Existing Thesauri

Undoubtedly the best known thesaurus is Peter Mark Roget's Thesaurus of English Words and Phrases, first published in 1852. The main use for Roget's Thesaurus is as an aid in English composition. Roget's Thesaurus links idea-words and text-words and hence the writer can use it as an aid in expressing ideas in text form. In the Introduction to Roget's Pocket Thesaurus [55], I.A. Richards contrasts a dictionary and the Thesaurus and gives a very good description of the use for Roget's Thesaurus:

You turn to a dictionary when you have a word but are not sure enough what it means -- how it has been used and what it may be expected to do. You turn to the Thesaurus when

you have your meaning already but don't yet have the word. It may be on the tip of your tongue, or in the back of your mind or the hollow of your thought, but what it is you don't know. ... But the word which just fits the bill won't come, so you reach for the Thesaurus.⁴

Roget's basic idea of using a thesaurus to pass from idea-words to text-words has been adopted by the originators of the majority of modern day thesauri. The ability to pass from one concept to another via indicated relationships is still the dominant aspect of most thesauri. As is illustrated below the difference between the Roget type thesaurus and the modern day thesaurus is the way in which this basic idea is used.

Most modern thesauri are "special purpose thesauri" in that they are used in particular fields by the people in these fields. A number of such thesauri will be discussed to illustrate those covering various subject areas.

A good example of a thesaurus applicable to a specialized discipline is the Water Resources Thesaurus [84] published by the Office of Water Resources Research of the U.S. Department of the Interior in November 1966. The Water Resources Thesaurus is a "vocabulary for indexing and retrieving the literature of water resources research and development". The thesaurus was developed to aid researchers and others in obtaining information on water resources problems. It was hoped that the thesaurus would be used by the authors of water resources literature to provide indexing terms for documentation purposes. The consequent improved information content of

4 Richards, I.A., Roget's Pocket Thesaurus, ed. C.O.S. Mawson, Richmond Hill, Ontario, Simon and Schuster of Canada, 1970, Page v, Introduction.

titles, abstracts and descriptors that results from the use of the Water Resources Thesaurus should aid in retrieval of information in this field. A similar statement could be applied to all thesauri used in special fields.

In 1969 H.N. Watts [85] compiled A Thesaurus for Health, Physical Education and Recreation. The thesaurus is used "in the tagging of documents for input into an information retrieval system for physical education"⁵. The thesaurus is also used in the preparation of search requests which are handled by the information retrieval system associated with the project. The intention is to insure that the terms used at the time of indexing are closely congruent to the words chosen to form search profiles.

The Thesaurus of Information Science Terminology [61] compiled by C.K. Schultz is a thesaurus used to index and search Professor Schultz's literature. The literature consists of approximately 3000 documents. Information Science students at the Drexel Institute of Technology have used this thesaurus "to guide their practice work in thesaurus-building, indexing and searching"⁶.

The Thesaurus of Pulp and Paper Terms [80] was compiled by the Pulp and Paper Institute of Canada and the Institute of Paper Chemistry. Through the use of this thesaurus it is felt that better "storage and retrieval of technical information within the area of

5 Watts, H.N., A Thesaurus for Health, Physical Education and Recreation, University of Alberta, 1969, Page 1, Preface.

6 Schultz, C.K., Thesaurus of Information Science Terminology, Washington, D.C., Communication Service Corporation, 1968, Page iii, Preface.

pulp and paper technology"⁷ will be possible. The purpose of the thesaurus is as follows:

It is, instead, an attempt to distinguish the more important concepts, within a particular area of interest, and to derive from these a list of authorized indexing terms, together with their associated cross-relationships. ..., the use of a common vocabulary by both indexer and searcher leads to improved efficiency of information retrieval.⁸

The Thesaurus of ERIC Descriptors [79] was compiled by the Education Resources Information Center Bureau of Research based in Washington, D.C. It is intended for use in storing, searching, and distribution of educational information.

The Thesaurus of Engineering and Scientific Terms [78] was prepared by the Engineers Joint Council and the Department of Defense. Printed in 1969 this is a thesaurus for both the Engineers Joint Council and the Department of Defense. The objectives of the coordinated efforts of the two groups were:

to produce a comprehensive thesaurus of scientific and technical terms for use as a basic reference in information storage and retrieval systems and to provide a vocabulary groundwork by means of which the interchange of information might be enhanced.⁹

The thesaurus is available on magnetic tape. It covers the areas that were previously covered by thesauri produced separately by the two groups. These were the Thesaurus of Engineering Terms and the Department of Defense Technical Thesaurus.

7 Thesaurus of Pulp and Paper Terms, Pointe Claire, Quebec, Pulp and Paper Research Institute of Canada, 1965, Page iii, Foreword.

8 Ibid, Page iii, Foreword.

9 Thesaurus of Engineering and Scientific Terms, New York, Engineers Joint Council, 1969, Page 1.

The paragraphs above indicate the varied areas in which thesauri are in use. In almost every field in which information is documented the use of a thesaurus leads to more efficient indexing and retrieval.

Further examples of thesauri will be mentioned later.

2.3 Thesaurus and Classification Linkage

As indicated in Chapter 1 and further illustrated in Section 2.6.5 there has been little discussion of the thesaurus/classification linkage, either manual or automated, in the literature. Papers by Davis and Aitchison provide the only indication of any work related directly to the approach taken in this thesis.

C.H. Davis [18] discusses the linking of vocabularies and classification schemes. Davis contends that the thesaurus entries by themselves are not as important as the thesaurus entries included with their associated relationships. He further states that vocabulary problems associated with thesauri (synonymy and term usage) can be largely eliminated by including classification codes along with the thesaurus entries. In the classification schedules the classification numbers would be followed by the corresponding terms. In application of this to a retrieval system the documents would be indexed by classification numbers instead of terms. Matching for retrieval would then be done on classification numbers. Davis suggests that the Universal Decimal Classification could be used for a scheme of this type. He feels that implementation of his ideas would place the user in more efficient and direct contact with the data

bases of concern to him.

The integration of a thesaurus with classification schedules along the same lines that Davis suggests has been carried out by the English Electric Company [3]. J. Aitchison [2] has described their scheme. A faceted classification scheme and a thesaurus covering engineering, scientific and technical, and management subjects have been integrated to form what is called the Thesaurofacet. The terms appear in both the schedules and in the thesaurus and are linked by the notation or class number. The thesaurus serves as an index to the classification schedules. The Thesaurofacet is a multipurpose tool; it can be applied to coordinate indexing, computerized retrieval systems, classification and shelving. Essentially the Thesaurofacet is a noncomputerized approach to the concept of linkage of a thesaurus and a classification scheme.

Several FID sponsored studies, not yet completed or published, indicate that some researchers are interested in further manual implementation of related ideas. Readers are referred to M.A. Mercier's thesis [48] for further references.

2.4 Description of Thesaurus Format and Relationships

Almost all thesauri developed since approximately 1960 are information retrieval thesauri. That is, they are used in the indexing and searching phases of an information retrieval system. This date was arrived at for two reasons: (1) J.C. Costello Jr. [17] considers this to be a significant date; (2) the author observed a similar trend during the literature review he conducted. Hereafter

these thesauri are referred to as "modern thesauri".

In the modern thesaurus terms are linked to other terms by indicated relationships. Examination of a page of almost any thesaurus will show that "main terms" are listed in alphabetical order. Beneath the so-called "main term" a series of "other terms" are usually listed. Normally these "other terms" are preceded by phrases such as "broader term", "narrower term", "related term", or by abbreviations which stand for these phrases. The series of phrases and "other terms" indicate the relationships of the "main term".

An explanation of some of the relationships in common use today follows:

- (1) scope note -- This is a short explanatory note intended to clear up any ambiguity in meaning or to define the use of the term.
- (2) use -- Appearance of this relationship usually indicates that the main term is not acceptable as an indexing or searching keyword and that the term following the "use" reference should be used in its place.
- (3) used for -- The term(s) following the "used for" reference are the terms which have been referred to the "main term" under which the "used for" reference appears. Obviously this is the reciprocal of the "use" reference. The "use" and "used for" references are illustrated below.

HAIL
USED FOR ICE STONES

ICE STONES
USE HAIL

- (4) narrower term -- The term(s) following this reference belong to

the class of concepts described by the "main term" under which they appear. These references give a more specific term associated with the "main term" and therefore facilitate more specific indexing or searching.

- (5) broader term -- The term(s) following this reference indicate the class of concepts to which the "main term" belongs. The references give a more general term associated with the "main term" and therefore allow for more general indexing or searching. This is the reciprocal of the "narrower term" reference. The "narrower term" and "broader term" references are illustrated below.

```
HAIL
  BROADER TERM  ICE
```

```
ICE
  NARROWER TERM  HAIL
```

- (6) related term -- The term(s) following this reference are related in some way to the "main term" under which they are listed. Essentially this is a means of broadening the scope of either indexing or searching. The "related term" reference has itself as a reciprocal relationship. The "related term" reference is illustrated below.

```
FREEZING
  RELATED TERM  HAIL
```

```
HAIL
  RELATED TERM  FREEZING
```

- (7) synonym -- The term(s) following this reference are term(s) which are synonymous with the "main term" under which they are listed. This reference has itself as a reciprocal relationship.

In many thesauri "synonym" references are not indicated. Instead synonyms or near synonyms have a "use" reference (explained previously) which gives the user the preferred term to use in indexing or searching.

These are some of the relationships that are indicated in most thesauri. Obviously some thesauri will include relationships which were not explained here. However, the ones explained are the most common relationships.

2.5 A Thesaurus and Information Retrieval Effectiveness

The effectiveness of an information retrieval system should be its most important characteristic. The standard means of evaluating an information retrieval system will be discussed with the hope that in light of this discussion the reader will realize that vocabulary control in the form of a thesaurus helps to improve system effectiveness.

G. Salton [58] feels that system evaluation first involves consideration of the different types of evaluation environments. According to Salton some of the factors comprising the evaluation environment are: type of user, request rate, document collection, type of indexers, type of searching, accuracy of search, response time, cost, and time lag. Furthermore, Salton feels that when a system's efficiency is being judged one must adopt a viewpoint which has most importance in the evaluation environment.

C. Cleverdon of the College of Aeronautics in Cranfield England has been responsible for much of the significant work in

this area. As quoted by Salton [58], he feels that six criteria affect user satisfaction: (1) coverage of the document collection, ie. does the system include relevant material; (2) time lag -- period between the time a search request is made and the time an answer is given; (3) form of presentation of output; (4) the effort required by a user to obtain answers to a request for information; (5) the recall of the system -- the proportion of relevant documents which are retrieved in searching; and (6) the precision of the system -- the proportion of retrieved documents which are relevant to the question.

Cleverdon contends that the coverage, time, presentation, and effort are easy to assess. He feels that system characteristics can be altered to overcome any dissatisfaction caused by any of these four factors. He states that recall and precision are difficult to evaluate and that alteration of these factors is quite difficult when the performance of a system falters.

One important ever-present factor is the cost of operations. Naturally this factor is very important when considering the effectiveness of any system. Salton feels that costs are handled by the user satisfaction criterion. Specifically, he feels that offering various classes of service at different costs partially accounts for the cost factor. H.S. Heaps and L.H. Thiel [32] have computed cost functions for searches conducted on Chemical Titles tapes at the University of Alberta. Their conclusions regarding cost of information retrieval are similar to those of Salton in that they feel that different services should be subject to different charges.

In an information storage and retrieval system the criterion

of effectiveness should be the ability of the system to satisfy the user at a price the user is willing to pay. This means that the primary aim of a system is to satisfy the information needs of its users.

Salton and Cleverdon feel that the main factors that should be used in estimating the cost of a system are recall and precision, which are closely related to accurate word usage.

Cleverdon's results regarding recall and precision in relation to word usage in an information retrieval system are as follows. He found that precision is determined by the specificity of the keywording. He also found that recall is determined by the exhaustivity or depth of keywording. If a person preparing questions for an information retrieval system keeps the aforementioned criteria in mind he can formulate his questions to obtain either high recall or high precision. Furthermore, by employing vocabulary control in the form of a thesaurus, the user will be more assured of having some sort of control over retrieval effectiveness as measured by recall and precision.

Quotations from two of the many articles [31, 33, 39, 40, 41, 58, 73, 76] which mention recall, precision, and word usage, should give the reader some idea of the use of recall and precision in evaluating information retrieval systems. F.W. Lancaster [39] says:

To measure the ability of the system to let through wanted documents and to hold back unwanted ones we must consider recall and precision jointly.¹⁰

S. Herner, F.W. Lancaster, and W.F. Johanningsmeier [33] say the

¹⁰ Lancaster, F.W., "Evaluating the Small Information Retrieval System", Journal of Chemical Documentation, Volume 6, 1966, Page 158.

following about the usefulness of relevance and recall ratios:

They serve as indices against which the effects of system or operational changes can be measured. A single set of relevance and recall ratios, backed by detailed analysis of causative factors, can give us a general picture of what a system is doing and some indication of why it is doing it.¹¹

The understanding of recall and precision should bring the reader to conclude that retrieval effectiveness could be increased by using a thesaurus for vocabulary control. In a computerized information storage and retrieval system this effectiveness could be still further enhanced by allowing the user to manipulate a thesaurus on-line through a terminal connected to a computer. The user sitting at a terminal, adding to, modifying, or displaying a thesaurus or parts thereof, in a man-machine interactive environment, would make use of the storage and logic capabilities of the computer and the intelligence of the human.

2.6 Literature Concerned with Thesaurus Concept

2.6.1 Introduction

Any newly developed significant and interesting idea that has rapidly risen from relative obscurity to a position of importance has resulted in a large and rapid increase in the literature. The expansion in the overall volume of written material to cope with the flood of new ideas has necessitated that efforts be directed towards achieving effective control.

¹¹ Herner, S., Lancaster, F.W., Johanningsmeier, W.F., "A Case Study in the Application of Cranfield System Evaluation Techniques", Journal of Chemical Documentation, Volume 5, 1965, Page 95.

The idea of using a thesaurus as a control in information retrieval does not appear to have received widespread attention until the late 1950's. T. Joyce and R.M. Needham [38] explain the problems involved with retrieval in systems that employ classification. They also describe some of the hazards involved in indexing without some form of synonym control. They claim that synonym control, especially in systems that employ a large number of terms in indexing, can be accomplished by using the thesaurus approach. B.C. Vickery [82] contends that to aid the indexer and searcher in information retrieval it would be useful to have some means of linking text-words and search words with standard keywords. He states that the means of accomplishing this aim might be a thesaurus. These two articles typify the starting of a new found interest in the thesaurus concept.

The literature that follows these early articles appears to consider the thesaurus concept from one or more of the following five viewpoints: (1) how to construct a thesaurus; (2) listing of an existing thesaurus; (3) description of how a particular thesaurus is used in an existing information system; (4) the use of a thesaurus as an aid in indexing, searching, and classifying; and (5) reviews of existing thesauri and thesauri literature.

Some of the articles cited will be mentioned more than once. The reason for this duplication is that the authors treat the thesaurus concept as being important from more than one of the above specified viewpoints.

2.6.2 Literature Dealing with Thesaurus Construction

A large part of the literature concerned with the thesaurus concept is devoted to the construction of a thesaurus. This portion of the literature usually mentions how a particular thesaurus was built or how a thesaurus might be built.

An article by J.F. Blagden [9] centers on how a thesaurus can be compiled. This article is described in greater detail in Section 2.6.6. This article crosses the artificial boundaries established in the introduction to this section in that it is an article concerned with compilation methods and it is also a review article.

E. Wall [83] gives a description of why vocabulary should be controlled and how this control can be achieved. He contends that various terminological conventions should be set before candidate term assembly takes place. He recognizes that a thesaurus will never be complete and devotes part of his article to the updating procedures that should be employed.

R.M. Rostron [57] describes how his architectural firm constructed a thesaurus that was to be used in the indexing and retrieval of technical literature in the field. He states that initially classification was considered as the tool to use in retrieving technical information but this approach was rejected for two main reasons: (1) they felt that the two major classifications for building lagged behind technical advance and current thinking in building; and (2) they felt that retrieval might become less precise if each item of information was cate-

gorized by fitting it into a limited number of classes. The task of constructing a thesaurus was then undertaken. First various aims were formulated based on shortcomings of already existing thesauri. Then a comprehensive word list was prepared. This article illustrates the way in which a group undertook the task of constructing a thesaurus while lacking previous experience.

D.F. Hersey and W. Hammond [34] describe how a computer was employed in the development of a Water Resources Thesaurus. Their article explains why a thesaurus was deemed necessary, the problems encountered, and the methods used in carrying out compilation with emphasis on the computer techniques used. The terms present in the thesaurus were chosen mainly from the United States Science Information Exchange (SIE) word lists and from terms used in indexing information in the Water Resources Research Catalog. The terms eventually chosen for inclusion in the thesaurus were chosen by a committee (construction by experts method of thesaurus construction -- see Section 3.2). Basically the computer was used to:

- (1) edit data format;
- (2) check for spelling consistency;
- (3) generate reciprocal entries for relationship entries;
- (4) generate "generic trees".

All of the terms to be entered into the thesaurus were keypunched according to a specified format dependent on the usage of the term. Hersey and Hammond claim that the computer techniques used in the compilation of the thesaurus are the first ones disclosed to the open literature. From the literature review conducted by the author their claim appears to be legitimate.

A.N. Grosch [30] describes a method by which a nonscientific thesaurus might be constructed. This article is directed towards the librarian but it is a good general article because it explains the "why" and "how" aspects of thesaurus construction. Some of the more important ideas mentioned are: (1) the importance of the specification of interrelationships among terms and the relationships that should be specified; (2) the need for a set of rules for grammatical structure of terms; and (3) the importance of updating a thesaurus.

J.L. Eller and R.L. Panek [24] describe how a thesaurus was developed for a decentralized information network. This article describes the development of the Education Resources Information Center (ERIC) Thesaurus mentioned previously in Section 2.2. This article describes how the project developed from its initialization in September 1965. "Free indexing" guidelines were prepared because of lack of a thesaurus. From the terms used in indexing documents a thesaurus base was created. The panel in charge of thesaurus development then developed a set of rules and conventions to apply to the thesaurus. They felt that already existing rules for thesaurus preparation (EJC and Project LEX) were inadequate in the field of education and for many of the social sciences. Any changes made to the thesaurus had to adhere to these rules. This article illustrates the way in which a thesaurus was developed in a decentralized environment. Because of increasing efforts towards national information exchange with information centers located in various cities the work done by ERIC may have widespread influence.

S.J. Martinez, L.P. Brown, D.P. Helander, and H.O. McLeod [45]

describe how a computer was used in generating the Exploration and Production Thesaurus, which is used in indexing Petroleum Abstracts. Initially a computer was used in printing the final copy of the thesaurus. Eventually programs were written that perform data manipulations which result in the eventual listing. New terms and relationships are keypunched according to a specified format. In this instance term selection is an intellectual process. The programs automatically generate the cross-references "narrower term" and "used for" for the "broader term" and "use" relationships, respectively. This article can be used as a guideline for the preparation of a thesaurus with computer assistance.

M. Wolff-Terroine, N. Simon, and D. Rimbert [86] describe the methods used in building a trilingual thesaurus (English, French, German) on cancer for use in an international information system. A list of indexing terms was used as the starting data base. Committees assigned to certain areas of the field were set up. They were responsible for establishing vocabularies and links among the terms within their particular area of concern. The resulting information was punched on computer cards and through use of computer programs the thesaurus was compiled. Computer programs were then written which performed the following functions: (1) input of new data; (2) elimination of data; (3) replacement of data; (4) generation of reciprocal relationships. This article illustrates a way in which computerization was used in compiling and maintaining a thesaurus.

The introduction or preface sections of most bound thesauri indicate the methods used in their compilation. References [77, 78,

79, 80] are some of the thesauri that indicate their method of construction in the introduction or preface sections.

In addition, articles by Holm and Rasmussen [35], Salton [58], Mandersloot, Douglas, and Spicer [44], Sparck Jones [68], Rolling [56], and Oller [50] are all concerned with thesaurus construction.

A majority of the articles concerned with the thesaurus concept mention something about thesaurus construction. The articles cited in this section give the reader a reasonable cross-section of the published material on this phase.

2.6.3 Actual Thesauri

Another large part of the literature concerned with the thesaurus concept is provided by thesauri themselves. A number of thesauri have been mentioned previously (see Section 2.2). As is evident from the examples given a particular thesaurus might be concerned with the terminology in a given discipline or it might serve as an aid in the indexing and searching functions applied to a document collection. A thesaurus can exist for the vocabulary of almost any discipline. Some additional thesauri which illustrate the wide-spread applicability of the thesaurus concept are: Euratom Thesaurus [25], American Petroleum Institute Information Retrieval System Subject Authority List [37], the United States Department of Agriculture Agricultural/Biological Vocabulary [1], the Case Western Reserve Thesaurus of Education Terms [7], and the Armed Services Technical Information Agency* Thesaurus of ASTIA Descriptors [77].

* now Defense Documentation Center

The particular thesaurus will almost certainly contain other things besides the list of terms. Included among these other things might be: a discussion of what prompted development of the particular thesaurus, what the thesaurus is used for, how the thesaurus was compiled, how the thesaurus is updated, and rules for term selection (singular-plural forms, noun forms, etc.).

The Euratom Thesaurus [25] gives the best display of concepts and relationships. Through use of the terminological charts indexing and query formulation become simple procedures. Page-thumbing to obtain necessary terms is largely eliminated. It should be noted that the display of thesaurus entries and relationships as found in this thesis is an adaptation of the terminological chart idea.

Scientists and engineers from 1960 to 1970 have learned to use thesauri such as those described above because of promotional efforts within their societies. Therefore this section of the thesaurus literature may be the best known.

2.6.4 Use of a Thesaurus in an Information System

Another portion of the literature concerned with the thesaurus concept centers on the way in which a particular thesaurus is being used in an information retrieval system. There are not many articles concerned with this phase of the concept even though it is probably foremost in importance. One reason for this might be the fact that the majority of information systems that use this type of approach are based in industry and industry frequently withholds information

and ideas that might possibly be of value to rivals.

In the most sophisticated information retrieval systems the use of a thesaurus may be very significant. Discussion of certain articles will illustrate the position of the thesaurus in various systems.

J.F. Caponio and T.L. Gillum [12] describe how the United States Department of Defense has attempted to solve some of the problems inherent in the information explosion. The authors quite correctly contend that before the results of research can be used effectively they must be made available to everyone who can use them. The Armed Services Technical Information Agency (ASTIA)* has been responsible for the collection, analysis, and dissemination of the research findings sponsored by the Department of Defense. ASTIA provided a service to DOD contractors and military organizations by supplying them with wanted information in the form of reports and bibliographies. In an environment where the report collection is increasing by approximately 30,000 reports per year efficient techniques for handling both incoming information and requests for information are necessary. The ASTIA Thesaurus [77] was developed as a tool to provide a congruence between the indexing and searching functions of the Agency. Automatic data processing techniques are used to perform literature searches and process report requests. Punched cards containing document accession numbers are the machine output from these procedures. The need for revision of the thesaurus is made by examination of word associations in input and output. The authors claim:

* now Defense Documentation Center

The utility of this concept as a tool for storing and retrieving technical report literature and for formulating accession indexes has been demonstrated by successful use at ASTIA over a period of many months.¹²

H.G. Sommar and D.E. Dennis [67] describe a system in use at the Celanese Fibers Company in Charlotte, North Carolina. The Celanese Fibers Company established an information retrieval system based on weighted term searching in January 1968. Because the document collection involved was unique to the Company, existing textile thesauri were found to be inadequate so a thesaurus was specifically developed for the document collection with which the retrieval system operates. The thesaurus, called the Thesaurus of Manmade Fibers and Textile Terms, is structured in the usual fashion so that the associated list following each indexing term includes narrower terms, broader terms, and related terms and this structure is made use of in searching. For example, in a search question put to the retrieval system each term in the question is assigned a weight. Then the computer expands the strategy by including from the thesaurus all of the narrower terms belonging to the terms specified in the question. The expansion of search strategy to include narrower terms is similar to the expansion of search strategy idea used by MEDLARS. Every document in the collection that is indexed by a term in the "expanded" list of terms earns a value equal to the weight assigned to the term. After all terms in the search question have been considered the total accumulated weights for each document are considered. Documents for

¹² Caponio, J.F., Gillum, T.L., "Practical Aspects Concerning the Development and Use of ASTIA's Thesaurus in Information Retrieval", Journal of Chemical Documentation, Volume 4, 1964, Page 8.

which the preset "minimum score" or threshold value is met or exceeded by this total are deemed "hits" for that question. The authors claim that this method of searching has many advantages. They say that the number of search terms required is reduced, that the search is more thorough, and that the automatic inclusion of narrower terms allows for searching at any level of specificity. The thesaurus is updated twice a year to allow for the inclusion of new terms. Only the indexing terms and their narrower terms are stored in the computer because only narrower terms are automatically included in any search strategy.

P.J. Horvath, A.Y. Chamis, R.F. Carroll, and J. Dlugos [36] describe an information retrieval system used by the B.F. Goodrich Company which incorporated three elements (namely a thesaurus, a dual dictionary, and a dissemination system) that they state are common to many information systems. The system uses the Engineering Joint Council (EJC) method of indexing.

The thesaurus used in the system consists of six sections. The first two sections are equivalent to the Chemical Engineering Thesaurus. The next four sections consist of a personnel list, a company list, project numbers and B.F. Goodrich Products. The interesting thing to note here is the fact that they did not attempt initially to create a thesaurus containing all the terms they felt belonged in it. Instead the terms used in indexing were considered as possible thesaurus entries by a group of persons who considered various criteria in evaluating terms for possible inclusion in the thesaurus. The thesaurus is stored on magnetic tape. Various codes

are assigned to incoming terms depending on their alphabetic position in the list of terms (term code), their relationship to other terms in the thesaurus (cluster code and use code), the section in the thesaurus into which the term is to be entered (section code), and the action to be taken with the term (deletion code). The existing thesaurus is updated by sort/merge procedures using the existing thesaurus tape and the incoming term information (on punched cards). Magnetic tape was chosen as the storage medium for printout purposes, updating purposes, and for use in the computer preparation of the dual dictionary.

Part of the complete thesaurus called the Short Thesaurus consisting of main terms and certain relationships is used in compilation of the dual dictionary. The dual dictionary is a list of index terms showing the accession number of each document containing the index term. After a document is indexed a card is punched for each term used in indexing the document, containing among other things, the term, the role of the term, and the document accession number. The Short Thesaurus is used for validation of these terms, generic expansion, synonym exchange (abbreviations, singular and plural forms, erroneously entered terms). A tape prepared from this step is merged with the old dual dictionary tape to produce the new dual dictionary tape. The dual dictionary is used in performing literature searches via manual coordination of terms.

The Automatic Information Distribution (AID) section of the B.F. Goodrich information system allows users to obtain document numbers which are retrieved via a weighted term search using

the index term cards (used in preparation of the dual dictionary) and user prepared profiles. The terms in the user prepared profiles are then accessed in the thesaurus to obtain the numeric code previously assigned to the term. Profile term cards are then punched containing this code and other information. These cards serve as input to the weighted term search procedure in the Automatic Information Distribution section of the information system.

The ways in which the thesaurus is used in this application are particularly interesting because they illustrate that a thesaurus can be used for more than the indexing and searching functions in an information retrieval system. The approach of this thesis is similar to that of the B.F. Goodrich information system in that in both instances utilization of the thesaurus is expanded.

M.M. Eichhorn and R.D. Reinecke [23] describe how a thesaurus serves as an integrating unit in a computer-based information storage and retrieval system at the Vision Information Center. The Vision Information Center (VIC) is concerned with, among other things, the organization and maintenance of records containing new information on ophthalmology and the visual sciences. A thesaurus is the key to the system. The thesaurus is made up of terms used in indexing documents which make up the bibliographic data base handled by the center. Computer programs handle the addition and deletion of thesaurus entries, and listing of the thesaurus in more than one form. Codes are assigned to terms upon entry into the thesaurus. These codes are used in both the indexing and

searching phases of the system. When searching, articles are returned as "hits" only if all of the thesaurus entries or codes specified in the question are used in indexing the document.

In addition, articles by Schultz, Schwartz, and Steinberg [62], Salton [58], Robinson [54], and Clough and Bramwell [14] all describe the use of thesauri in information storage and retrieval systems.

2.6.5 Value of a Thesaurus as an Aid in Indexing, Retrieval, and Classification

Without an exception all of the literature concerned with the thesaurus concept mentions the value of a thesaurus as an aid in indexing and retrieval. The authors are unanimous in agreeing that this is the reason for the existence of thesauri in information systems.

In Section 2.5 articles were cited which mention in a general fashion that the measures of retrieval effectiveness, recall and precision, would be improved by employing a thesaurus in system operation. An article by C.W. Hargrave and E. Wall [31] gives an account of a controlled experiment to measure effectiveness. Their work dealt with the National Aeronautics and Space Administration (NASA) experimental Selective Dissemination of Information system. They tested recall and precision for six months prior to the introduction of the NASA Thesaurus (as an indexing and searching guide) and compared the results with those obtained during the six months that followed the introduction of the thesaurus. Their tests showed that after introduction of the

thesaurus precision was reduced slightly and recall was improved greatly. Appropriately, they conclude their article with the statement: "... the utility of the Thesaurus is clearly demonstrated"¹³.

Of particular interest to this thesis are the articles concerned with the value of a thesaurus as an aid in indexing and retrieval where there is unification of the thesaurus approach and classification techniques. As mentioned previously J. Aitchison [2] describes the integration of classification schedules and a thesaurus. The article is concerned with the English Electric Thesaurofacet [3] which is a faceted classification scheme and thesaurus covering fields in science and technology. Terms in the schedules appear in appropriate facets and hierarchies. The thesaurus gives other possible hierarchies and relationships which allow one to enter the classification schedules in a different place. The thesaurus is also an alphabetical index to the class numbers. The terms appear both in the thesaurus and in the classification schedules. The notation or class number is the link between the term in the thesaurus and the term in the schedules. The Thesaurofacet can be used for classifying books and as a controlled language tool for indexing and searching. The author makes this claim about the Thesaurofacet:

... by uniting a thesaurus approach with classificatory techniques, it stands to give advantage in the display of concept interrelationships which might not be

13 Hargrave, C.W., Wall, E., "Retrieval Improvement Effected by Use of a Thesaurus", Proceedings of the American Society for Information Science, Volume 7, 1970, Page 293.

achieved by either method alone.¹⁴

Nothing was mentioned about automation of the Thesaurofacet.

Other articles which are concerned with the integration of a thesaurus and classification schemes have been written by London [42], Davis [18], and Sparck Jones [69].

2.6.6 Thesaurus Reviews

Some of the thesaurus literature consists of reviews of existing thesauri or reviews of thesauri literature. The references in this area of the thesauri literature are very useful. Each article cited, by itself, can very adequately serve as a starting point for research into the thesaurus concept.

J.F. Blagden [9], in an article already cited, lists 70 references which cover most of the work done involving thesauri up to the time of publication of the article. He conducted an extensive investigation into the techniques of thesaurus construction before undertaking construction of a thesaurus of management terms. The article mentions many of the techniques advocated by various authors and in many cases discusses both advantages and disadvantages of the respective methods. He concludes that no one method can be used in the construction of a thesaurus. He explains the method he proposes to use in the construction of his thesaurus of management terms; it is made up of what he calls an "amalgam" of all the approaches he considered. This excellent article has an

¹⁴ Aitchison, J., "The Thesaurofacet: A Multipurpose Retrieval Language Tool", Journal of Documentation, Volume 26, Number 3 (September 1970), Page 203.

extensive bibliography and covers the area of thesaurus construction in some detail.

As a term project in Computing Science 560 at the University of Alberta in 1969-1970 F.J.J. Colgan [15] compiled a list of references which he stated covers the literature development in thesaurus work during the previous two years. He lists between 35 and 40 references and in most cases gives a one or two sentence description of each. This gives adequate coverage of the period in review and some excellent explanations of purposes. Certain of the articles Colgan has referenced appear as references in the thesis bibliography.

The Bibliographic Systems Center (BSC) at Case Western Reserve University, a center for library and information science, serves as a clearinghouse for classification schemes and thesauri. The center is especially concerned with the problems of handling scientific and technical information.

K. Gaster [28] lists approximately 80 references concerned with thesaurus construction and use that were in the ASLIB Library as of July 1967. She also lists 21 thesauri present in the ASLIB Library for reference purposes. Updates to this bibliography are available to interested parties. Some of the references are in French and some are in German. This, with Blagden's bibliography, covers a good portion of thesauri literature up to 1968.

The 1966 issue of the Bulletin De L'Association Internationale Des Documentalistes [63] is an excellent reference. Some articles in this publication are in French and some are in German

with the majority in English. For the most part the articles describe existing thesauri. Some of the thesauri described are: The Engineering Index Thesaurus, the Bureau of Reclamation Thesaurus of Descriptors, the Bureau of SHIPS Thesaurus [10], the Thesaurus of Pulp and Paper Research Terms [80], the Euratom Thesaurus [25], and the EJC Thesaurus [78]. Some of the thesauri described are mentioned in Sections 2.2 and 2.6.3. This publication is very useful because in total the articles in it: (1) describe various thesauri; (2) tell how these thesauri were compiled; (3) describe their use; (4) generally describe thesauri and the advantages and disadvantages associated with their use.

2.6.7 Conclusions

As already stated almost all of the literature surveyed considered the thesaurus concept from one of the six mentioned viewpoints. Some of the articles surveyed, however, looked at thesauri from different and unusual viewpoints. C.J. Surace [71] is concerned with different ways of displaying thesaurus entries and relationships. Surace is concerned with the utility of the different display formats for the indexer and searcher. H.H. Neville [49] has devised a scheme whereby the keywords in a thesaurus can be converted into the appropriate keywords of another thesaurus in the same subject area. Concepts are identified in each thesaurus and are assigned codes. These codes enable keywords in one thesaurus to be converted into keywords in another thesaurus.

By referring to the bibliography it should be obvious that

there is much information on the thesaurus concept available to the searcher. Nevertheless, the author feels that there is much unpublished information and that some of the most interesting projects are not discussed in the open literature; these are very difficult to locate.

CHAPTER III

CREATION OF A THESAURUS

3.1 Introduction

Essentially there are three approaches to the problem of thesaurus construction. These three approaches will be described and advantages and disadvantages of each approach will be mentioned.

3.2 Construction by Experts

The manual "construction by experts" method of approach to thesaurus creation is still the most common method used today. In the creation of a thesaurus by this method the essential step involves the careful consideration of the utility of candidate terms for describing the discipline which the thesaurus attempts to either totally or partially encompass. This so-called "careful consideration" is done by human judgement. The methods of attack used in the creation of some thesauri constructed via this approach will now be described.

The Thesaurus of Information Science Terminology [61] was initially prepared by Claire K. Schultz. The aim of this thesaurus, as described previously, is to facilitate easy searching of the author's literature selection. The author has updated the thesaurus to accommodate changes in her document collection. Students from the Drexel Institute have also suggested changes to it. However, this thesaurus has essentially been compiled by one person.

The Water Resources Thesaurus [84] was:

prepared by qualified scientists who carefully processed lists of candidate terms to determine their general utility for describing water resources research and development efforts and to identify the semantic relationships among them.¹⁵

The terms consisted of those used in the preparation of the Catalog of Water Resources Research and those suggested by scientists, engineers, and other specialists.

The method used in compiling the Thesaurus of Pulp and Paper Terms [80] is an illustration of another way in which a thesaurus can be prepared. Using internal indexes at the Pulp and Paper Research Institute a first draft was compiled. More terms were included by references to bibliographic indexes. This draft thesaurus was then used to index abstracts from the Abstract Bulletin of the Institute of Paper Chemistry. This procedure generated additional keywords and also pointed out deficiencies in the original draft. A committee then looked at each term individually, considering its utility, ambiguity in meaning, and cross relationships with other terms. The method used in compiling the Thesaurus of Pulp and Paper Terms is a good one. By applying the draft thesaurus to the type of situation in which it would undoubtedly be used, the inadequacies of the draft were brought to light and changes were made before final printing took place.

Several major thesauri were produced by a series of steps which closely adhere to the steps followed in the compilation of the Thesaurus of Pulp and Paper Terms. The main difference is that

15 Water Resources Thesaurus, Washington, D.C., United States Department of the Interior Office of Water Resources Research, 1966, Page vii, Introduction.

these thesauri were initially published and used in indexing and searching. From the experience drawn from use of these thesauri changes were made and updated issues were published. Two thesauri which were prepared in this manner are the Thesaurus of Textile Terms (Second Edition), and the Thesaurus of ERIC Descriptors (First Edition) [79], a follow-up to the Thesaurus of ERIC Descriptors (Interim).

In 1965 the creation of a new Technical Thesaurus was deemed necessary by the United States Department of Defense. The Office of Naval Research (ONR) was assigned the task. ONR designated this mission as project LEX. Besides creation of the actual thesaurus one of the project requirements was to prepare a manual which indicated the method used in building the thesaurus. This manual is an excellent guide which could be used in building almost any technical thesaurus. This manual covers almost everything involved in thesaurus construction, including fundamental term rules, cross reference rules, and alphabetization rules. About 35 people participated in deliberations which resulted in the development of the manual. The thesaurus produced was called TEST (Thesaurus of Engineering and Scientific Terms). It was the result of efforts of not only the Department of Defense but also the Engineers Joint Council. The Engineers Joint Council had set out in 1965 to revise the first edition of the Thesaurus of Engineering Terms. Overlapping of the interests of the Engineers Joint Council and the Department of Defense resulted in a merging of the efforts of the two projects and as already stated resulted in the Thesaurus of Engineering and Scientific Terms [78].

There are two problems involved with this method of thesaurus construction. The majority of thesauri created using this method exist in printed form either as a book or as unbound pages. Periodically the thesaurus will require updating. This almost certainly means that major reindexing will be required and hence reprinting will be necessary. This reindexing and reprinting can be both time-consuming and expensive. The second criticism of this method is more fundamental. Can a small group of people speak for the whole group that they represent? The opinions and actions of a small group tend to follow those of the most dominant member of the group, even after very little exposure. Thus the bias of one committee member could play a very significant role. Although a committee set up to choose terms for a thesaurus would be very carefully chosen, the previously mentioned situations may well occur. Nevertheless, the advantages of using this approach should be obvious. If a representative cross-section of people consider the utility of candidate terms in theory the thesaurus should have no special bias. This method of thesaurus construction has been the one most often used over the years and undoubtedly in the future it will continue to be widely used.

Many thesauri compiled using this method are stored on magnetic tape. This allows for computer usage in printing and updating a thesaurus. A portion of the literature described in Chapter 2 describes some of the efforts in this direction. A logical progression would be for more of the process to be automated. Time-sharing computer systems make it feasible for the experts to use a

terminal connected to a computer in order to make entries in a thesaurus or to access entries already present in a thesaurus.

3.3 Construction by Machine

3.3.1 Introduction

A method used to construct a thesaurus which either totally or partially eliminates direct human involvement is by a procedure which works with the frequency of occurrence of terms. G. Salton [58] in his SMART system has done much work in this area. Some of the work is described below.

3.3.2 Fully Automatic Methods

Fully automatic methods of thesaurus construction involve looking at the co-occurrences of terms in documents which are assumed to be representative of the subject area the thesaurus is concerned with. The methods assume that term co-occurrences in sample documents indicate that the respective terms are similar or related.

Initially a frequency count of the terms making up the documents is carried out. From this procedure a "term-document" matrix

| | T ₁ | T ₂ | T ₃ | T ₄ | T ₅ | T ₆ | |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|------|
| D ₁ | 3 | 0 | 0 | 2 | 0 | 6 | |
| D ₂ | 0 | 0 | 1 | 3 | 2 | 0 | |
| D ₃ | 0 | 2 | 3 | 0 | 4 | 0 | |
| D ₄ | 1 | 2 | 1 | 0 | 3 | 1 | |

Fig. 3-1: Term-Document Matrix

is obtained. The ij -th element represents the number of times term j appears in document i .

Salton assumes that terms are related when they co-occur in the same context. In Fig. 3-1 terms T_1 and T_6 might be assigned to a common thesaurus category because they both occur in documents D_1 and D_4 . The common thesaurus categories or term associations can be represented by an association map in which branches between terms represent the term relations.

3.3.3 Semi-Automatic Methods

In semi-automatic methods for thesaurus construction subject experts define the meanings of terms to be included in the thesaurus. The basis for thesaurus entries is a word frequency list usually generated by automatic means. By human observation and the answering of certain questions the terms can be represented by a "property matrix" (see Fig. 3-2).

| | P_1 | P_2 | P_3 | P_4 | P_5 | P_6 | |
|-------|-------|-------|-------|-------|-------|-------|---|
| T_1 | 3 | 0 | 0 | 5 | 1 | 0 | |
| T_2 | 0 | 1 | 3 | 0 | 1 | 0 | $P_i=0$ property inapplicable |
| T_3 | 2 | 0 | 1 | 5 | 2 | 0 | $P_i=1$ property applies somewhat |
| T_4 | 1 | 0 | 1 | 3 | 2 | 0 | |
| T_5 | 2 | 1 | 2 | 2 | 2 | 0 | $P_i=2$ property applies more strongly |

Fig. 3-2: Term-Property Matrix

By referencing the term-property matrix identical rows can be combined into a single group. By eliminating certain properties other terms may be grouped together.

Thus the essential steps in this method of thesaurus construction are: (1) a word frequency list is prepared (usually by automatic methods); (2) the different word usages for each word to be included in the thesaurus are decided upon; (3) questions are prepared which serve as a means by which term groupings can be ascertained; and (4) the resulting property matrices are compared and words identified by like properties are combined into groups.

3.3.4 Hierarchy Formation

In almost all thesauri hierarchical term relationships are present. Hence it is desirable to be able to generate a hierarchy automatically or semi-automatically.

One way to construct the hierarchy automatically is based on the term-document or term-property matrix. Given two terms defined by property vectors the following cases are possible: (1) the terms are identified by different properties and are unrelated; (2) the terms are identified by the same properties with neither term dominating the other, and the terms are placed in the same group; (3) the terms are identified by the same properties, but the weights for term A are greater than the weights for term B; A is placed on a higher level in the hierarchy; (4) the terms are identified by the same properties, but B dominates A; B is placed on a higher level in the hierarchy.

By computing a similarity coefficient the similarity between two property vectors can be ascertained. A possible similarity measurement is given by the following expression:

$$c_{ij} = \frac{\sum_k \min(v_k^i, v_k^j)}{\sum_k v_k^i}$$

where c_{ij} is the similarity between term i and term j , and v^i and v^j are k -dimensional property vectors representing terms T_i and T_j .

When the two property vectors are identical $c_{ij}=1$ and when the two vectors have no common properties $c_{ij}=0$. If terms T_i and T_j are to be entered into the thesaurus in a particular hierarchical structure a certain value K for the similarity coefficient must be met. The following conditions might be applied: (1) if c_{ij} and c_{ji} are $< K$, then terms i and j are not related; (2) if c_{ij} and c_{ji} are $> K$, then terms i and j are synonymous; (3) if $c_{ij} < K$ and $c_{ji} > K$, then term i is on a higher level in the hierarchy than term j ; (4) if $c_{ij} > K$ and $c_{ji} < K$, then term j is on a higher level in the hierarchy than term i .

A term-term similarity matrix can be generated from the term property matrix, by calculating similarity coefficients using the previously given formula. The cutoff value K is applied to the elements of the term-term similarity matrix. Elements which exceed the cutoff value in value are assigned a value of 1 while those which do not exceed the cutoff value are assigned a value of 0 in a reduced matrix. The reduced matrix can be readily used to produce the resulting thesaurus hierarchy if its elements are applied to the algorithm given previously. The cutoff value K is the critical factor in determining the kinds of relationships. If $K=0$ all terms

will be synonymous. As K increases some hierarchical arrangements will appear and some will disappear. As K increases further most terms will become unrelated.

A method of hierarchy formation which is semi-automatic in nature employs the word frequency lists and the questions used for creation of the term-property matrix. The answers to the questions asked result in classes of word uses. By applying more questions to the words in these classes finer subdivisions result.

Another method of hierarchy formation is based on word use frequencies. The word frequency list and a set of questions are employed to determine the word uses to be included in the hierarchy. A two-way hierarchy is started by initially choosing the word use with the highest frequency say word T_k . One node will represent word T_k and all similar words while another node represents all words not related to T_k . Partitioning amongst word groups continues until the groups are small enough to be entered as hierarchical classes in the thesaurus.

3.3.5 Conclusions

The advantage of automatic or semi-automatic thesaurus construction methods is that the unconscious bias of individuals in choosing thesaurus entries and associated relationship entries is largely eliminated. However, the vocabulary bias of the authors of the supposedly representative document collection plays an important role in the eventual contents of the thesaurus. However, this can be overcome by careful selection of the document base. Time is

saved and thereby costs are decreased by employing automatic or semi-automatic thesaurus construction methods. Such automatic or semi-automatic methods of thesaurus construction, to a large extent, eliminate problems associated with the construction of thesauri by experts.

3.4 Construction by Users in a Man-Machine System

With the advent of time-shared computer facilities construction of a thesaurus by users in a man-machine type system became feasible. This method is of particular importance in this thesis. This method of construction involves a user who communicates with a computer, normally through a terminal-type device. Basically the user types commands and other information recognizable by a computer program designed and written to carry out actions which are wholly predetermined. These predetermined actions might allow the user to add new terms to the thesaurus, delete terms from the thesaurus, or display terms and relationships. The allowable actions are dependent on what is deemed necessary when initial program design takes place.

There are definite advantages to this method of thesaurus construction. The user has the advantage of being able to extend the thesaurus as he uses it. The problem of scanning many printed pages for a desired term is avoided by having the computer program display any specified term and relationships upon appearance of a specified command and the desired term. The problem of reprinting the thesaurus after updating takes place is avoided; the entire

thesaurus with all updates is stored on magnetic tape, disk, or drum. The computer program can be designed and written to generate automatically any reciprocal relationships associated with specified relationships; the time-consuming task of establishing reciprocal relationships manually is avoided.

There are also problems associated with the construction of thesauri using this method; these problems are of both man and machine type.

As indicated earlier the difficulty of determining a suitable query language appears in any discussion of this method of thesaurus construction. How much information must the computer program convey to the user before expecting a correct response from him? Must the user be prompted before each reply? How complex must the query language be? It is reasonable to assume that after the user becomes familiar with the requirements of the program regarding command specification, the query language should be able to undergo much simplification without endangering the correct functioning of the man-machine interaction involved in accessing and modifying the given thesaurus.

There are certain significant computer problems associated with this method of thesaurus construction. First, the space that can be allocated to any one user of a time-sharing system is limited. Thus the space required to account for a very large thesaurus might not fit into the space allocated. Should the computer program be designed and written to perform the minimum necessary or should a more complex program which allows for more sophisticated actions be

written? The tradeoff is resolved by assigning values to the respective criteria depending on their relative importance in the overall picture. Second, the time-sharing facility may not be available on a 24-hour basis. Therefore, the user is able to make use of an on-line thesaurus only when the computer is handling terminal tasks. Third, telecommunications problems are associated with any terminal-type environment. The problems associated with dial-up and error-rates, that are dependent on a number of factors, detract from the satisfaction of using an on-line facility. Finally, there are the human aspects of the problem. People may not want to use the machinery. After the novelty of initial use wears off, especially if difficulties have been encountered, users may revert to the method they previously used, which still accomplishes the task, even though it may not be as efficient as the on-line alternative.

As stated, some tentative work have been done with on-line thesauri at the University of Alberta. This work will be discussed in a general way.

In the summer of 1969 Research Assistant A.L.S. Wong [87] investigated the feasibility of thesaurus manipulations in an on-line environment. He wrote a program which allowed a user to manipulate a thesaurus while sitting at a computer terminal. The program was written in PL/1 and operated under the time-sharing system CP/CMS at the University of Alberta. The main objectives of the investigation were: (1) to investigate the feasibility of an on-line thesaurus program; (2) to develop a program which would carry out operations necessary to manipulate a thesaurus in an interactive environ-

ment so query languages and user reactions could be investigated.

Wong's thesaurus program allowed five types of relationships for any one term. The relationships allowed for were: related terms, broader terms, narrower terms, synonyms, and preferred terms. Scope notes were also allowed for a portion of the thesaurus entries. The program allowed for 1000 terms, 200 of which could have scope notes. The program kept track of thesaurus entries and their associated relationships by employing tables for the entries and scope notes, and vectors and matrices of pointers for the relationship information.

The file handling techniques Wong uses accomplish the desired end with a minimum of program complexity. Critical evaluation shows that in certain places the efficiency of his method is not good. One inefficiency is that thesaurus entries and scope notes were all allocated fixed lengths. The terms were assigned a length of 24 characters while the scope notes were assigned a length of 48 characters. Thus if a term was only 10 characters long 14 available characters went unused. The same situation occurred with scope notes.

However, Wong's work: (1) pointed out that thesaurus manipulation in an interactive environment is definitely feasible and attractive from a user's viewpoint; (2) showed that limitations imposed by computer facilities are of definite importance and deserve much consideration in any investigation of this type; (3) pointed out important facts regarding the query language employed by the program.

An extension to the work done by Wong was carried out by a group of students as a term project in Computing Science 560 in the

1969-1970 term. Their work consisted mainly of programming and demonstrating the program and its workings to class members to gain further information from a large user group. The programming consisted of correcting "bugs" in the program written by Wong.

R.C. Sohnle [66], one of the members of the group that worked on the thesaurus as a term project for Computing Science 560, wrote a new thesaurus program during the summer of 1970 working as a research assistant. His program, written in PL/1, is operative under the MTS time-sharing system presently in use at the University of Alberta. He was much more concerned with the efficiency of programming and file handling techniques.

Sohnle's program is very good for demonstrating an on-line thesaurus. The query language employed in the workings of the program is very comprehensive and improved as a result of previous tests. It is made very clear to the user what is expected from him. For demonstrating purposes, for new users, or class instruction, the query language employed is very good. However, as Atherton and Miller [5] contend in their article describing MOLDS, carrying on a conversation with a computer via a very involved query language becomes frustrating as the user learns the operations that the query language controls. What is desired is an optimum mixture of computer induced response and response formulated independently by the user. Perhaps the optimum mixture varies as the user or audience varies.

In Sohnle's program 3000 terms with an average length of 11 characters are allowed for. The length of all the terms combined should not exceed 32767 characters. Scope notes, related terms,

broader terms, narrower terms, synonyms, and preferred terms are the relationships allowed for.

Sohnle's program shows more sophistication than Wong's in the file structure, the query language, and programming technique. Considering the general framework of allocating specified areas for certain relationships, improvement upon Sohnle's method for handling entries and associated relationships is difficult. What is needed, however, is an efficient and economic file handling technique which eliminates the allowance of only specified relationships and furthermore eliminates the allowance of only a certain number of one kind of relationship. Essentially this means that a more general program is desired, one that would allow different users to specify different kinds of relationships if he wants to. The accomplishment of this increased generality is desired without greatly increasing program complexity.

During the 1970-1971 winter session further small refinements to Sohnle's program were carried out by E.P. Krawchuk and D.H. Stosky. Further tests were conducted with students in Computing Science 560 in 1970-1971. The work showed that a more simplified query language was desirable.

As mentioned previously any endeavour of this type is hampered by computer limitations. The space allocated to one user by the computing system and such variables as terminal response time and disk access time are, for the most part, beyond control of the user. One can only design a program to make optimal use of that which is available.

The design criteria and implementation details for the computer program which handles the thesaural and associated operations for this thesis project are described in the next chapter. It should be noted that all references to "THESAURI" will refer to the computer program written by the author unless otherwise specified.

CHAPTER IV

THESAURUS PROGRAM

4.1 Introduction

This chapter discusses the on-line construction of a thesaurus. The thesaurus is regarded as the central feature of an integrated retrieval system; and its design, programming, and query language were investigated so that a user may create, modify, or display a thesaurus or parts thereof and move from the thesaurus to search programs in a man-machine interactive environment.

The key requirements for such an on-line thesaurus program are the following: (1) fast access to terms and associated relationships; (2) an unlimited number of relationships of a certain type associated with a particular term; (3) generality in that a user is not tied to specific relationships; and (4) brevity of program to allow more computer memory to be devoted to the required accounting and storage of information. The thesaurus program developed for this thesis attempts to fulfill these requirements.

4.2 Theoretical Aspects of File Handling Techniques Used

4.2.1 Introduction

The key requirement in a computer program or programming system which involves the manipulation of large quantities of data is an efficient file organization technique. There are various methods of handling files in data processing applications. The file handling technique adopted for a specific application is

normally dependent on the assignment of dollar values to variables such as the importance of fast access to information in the file, the importance of being able to quickly update the file, and the importance of storage considerations.

Many applications utilize files which are both searched frequently for pertinent information and altered frequently with new information. In information retrieval an example of this situation is word frequency counts in text samples. This is in contrast to some applications in which the files may be subject to one of the following situations: (1) accessed frequently and altered infrequently; (2) accessed infrequently and altered frequently.

In any thesaurus program the information must be regarded as being both frequently accessed and frequently altered. However, the accession feature is of greater importance than the alteration feature. The reason for this is that once a thesaurus has been created information is more likely to be accessed than altered.

In the thesaurus program associated with this thesis three file organization techniques were used. The characteristics of these techniques will be described; then the method of application of these techniques to the thesaurus program will be discussed. It is hoped that the explanations will show that the type of file organization used is efficient in meeting the stipulated requirements.

4.2.2 Definitions

An "item" is the basic unit operated on in any data processing application. An "item" consists of two parts: (1) the "key" is used

to distinguish between items; and (2) the "function" is the part of the item which is not the key. Here the term "data processing" refers to the "totality of operations performed by a computer"¹⁶. The term "item" is synonymous with the term "record". A further definition which might help to clarify the meaning of "item" or "record" is: "a group of related facts or fields of information treated as a unit, frequently consisting of organized or processed information"¹⁷. These definitions have been included because people searching the literature will quite often encounter the terms without their being defined.

4.2.3 Binary Search

The binary search technique is an efficient method of handling files which are searched frequently and altered infrequently. In order to facilitate use of the binary search technique on a file of N items the items must be arranged in such a way that their keys are in ascending or descending order.

The binary search method bisects the file until the desired item is located. The first test is made on the item at the midpoint of the file. The comparison made determines whether or not this is the desired item. If the item tested is not the desired item the comparison has determined in which half of the file the desired item is found. The bisecting of the effective file continues until the desired item is located.

The maximum number of comparisons needed to locate an item in

16 Fritz, W.B., "Selected Definitions", Communications of the ACM, Volume 6, Number 4 (April 1963), Page 155.

17 Ibid, Page 157.

a file of N items is $\log_2(N + 1)$. In a file of 255 items the maximum number of comparisons needed to locate any item would be 8.

An algorithm for a binary type search is given in Fig. 4-1. In the algorithm: "N" refers to the number of items in the file, "term" refers to the key of an item in the file, and "key" refers to the key of the desired item ie. the one we are looking for.

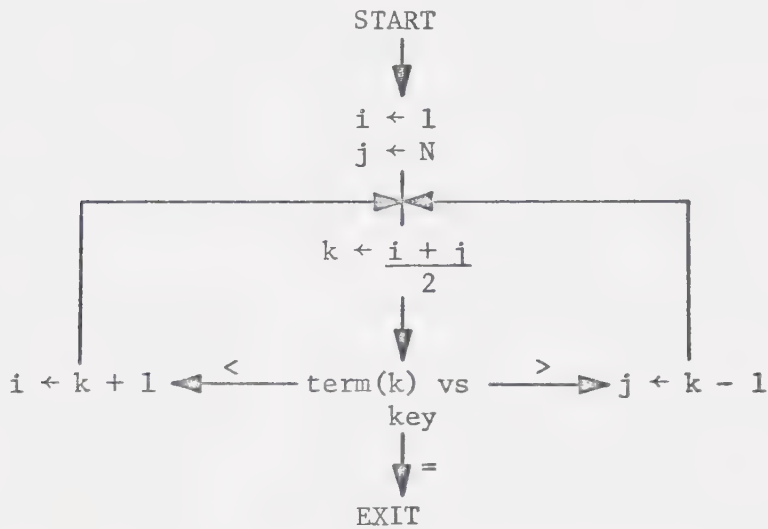


Fig. 4-1: Algorithm for Binary Search

The alteration of a file designed for a binary type search is time-consuming because the file items must be in ascending or descending order of key. This requirement means that many items might have to be moved either to make room for an incoming item or to delete an existing item.

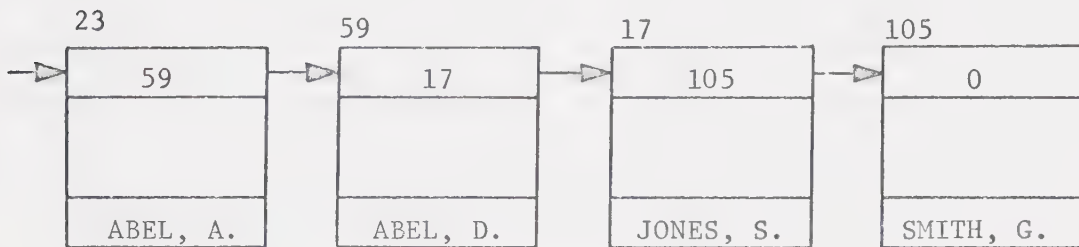
4.2.4 Chaining

The previous discussion shows that the limiting factor in using the binary search is the problem of altering the existing file. By using a technique called "chaining" (sometimes called "linked

linear list") the time required to alter a file can be reduced.

In this technique a "pointer" is included in each item or record. It "points" to the location of another item or record in the file. The pointer technique allows the logical and physical arrangements of items or records to be different.

The chained or linked structure idea is illustrated below.



The four items are in logical order (alphabetical order by name) but they are not in the same physical order. The logical order is obtained by the pointers within the items. The last item contains 0 in the pointer field indicating that this is the last item in the list.

To add a new item to the file the chain can be broken at any point and relinked with the newly inserted item. Deletion of an item from the file is also very easily accomplished.

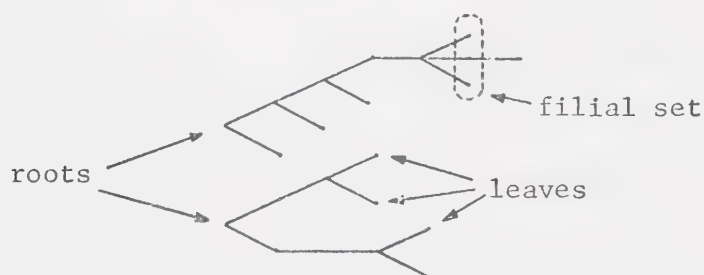
Obviously alteration of a given file is more efficiently accomplished by the use of the chained technique than by the binary search technique. The disadvantage of the chained file is noticeable when searching for items. Only one item is accessible from any other item and hence a chained file must be searched serially.

The advantages and disadvantages of the above two methods lead to their being used in situations where the advantages of their

respective usage play the more important role.

4.2.5 Tree Allocation

E.H. Sussenguth Jr. [72] has devised a scheme that efficiently handles files which are both accessed frequently and altered frequently. In order to understand Sussenguth's file processing procedure an understanding of a few definitions is required. The most important definition he gives is that for filial set: "the filial set of node x is the set of nodes which lie at the end of a path of length one from node x ".¹⁸ Most of the definitions important to understanding Sussenguth's technique are illustrated below.



Sussenguth gives a further definition which may help to clarify the concept filial set. "The filial set of each K th level node is comprised of those nodes which correspond to those elements actually used in combination with the element associated with the parent node."¹⁹ If the keys are English words the nodes might correspond to letters. Then the filial set for the letter B, for example, would be the letters that can be used with B to start a word.

¹⁸ Sussenguth, E.H. Jr., "Use of Tree Structures for Processing Files", Communications of the ACM, Volume 6, Number 5 (May 1963), Page 273.

¹⁹ Ibid, Page 274.

Fig. 4-2 gives a file and its associated tree structure.

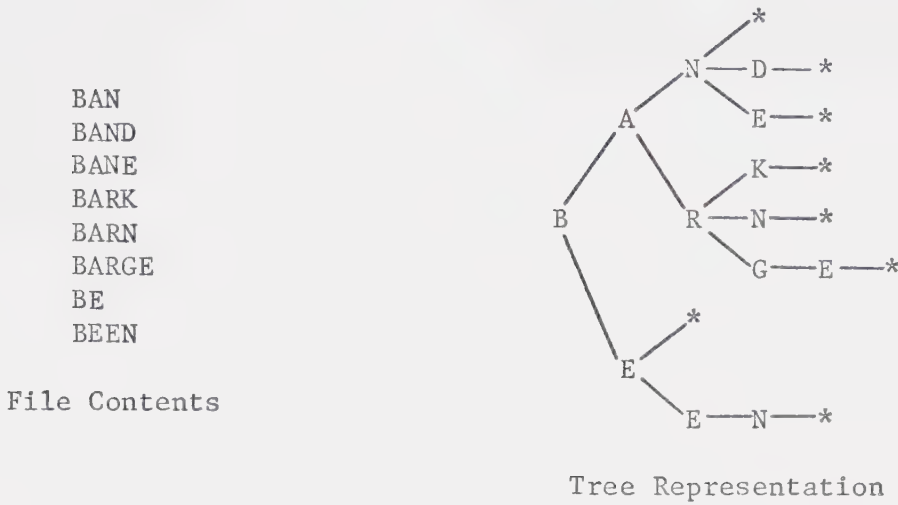


Fig. 4-2: File and Associated Tree Structure

A partial key can be associated with each node. The partial key would be made up of the node values from the root to the given node. This means that the key for a leaf is the key of the item to which the leaf corresponds. The partial key concept presents an interesting point. In the example given in Fig. 4-2 BAN is a partial key with respect to BAND and BANE and it is also a file item so it must be a leaf. By terminating each key with a special character (* in Fig. 4-2) file items correspond to leaves.

Searching the tree allocation for a given item is a simple procedure. Initially the roots are scanned to find the root corresponding to the first element of the key of the desired item. After the root is located the filial set for that root is accessed. The filial set is searched for the second element of the key of the desired item. Searching the filial sets for the elements making up the keys continues until the desired leaf is located.

Essentially the same procedure is used to add an item to the file. The file is entered as if a search were being undertaken. At some level a filial set is found that does not contain a node value that matches one of the elements of the key being added. At this point the filial set is expanded by including as a node the element of the key which previously was not a member of the filial set. The filial set of this newly added node consists of the next element of the key being added to the file. Additional filial sets are added until there are no more elements in the key.

Representation of Sussenguth's technique in a computer can be done in more than one way. One possible method is to chain all nodes to their filial sets and to chain the nodes within the filial set together. This technique is called "double-chaining". In terms of computer storage one portion of a word might contain the node value, a second portion might contain the address of another node in the same filial set of which the node being considered is part, and a third portion might contain the address of the first node in the filial set of the node being considered. Fig. 4-3 gives a pictorial representation of this concept while Fig. 4-4 gives the computer memory contents for the tree in Fig. 4-2. The double-chaining technique can make use of any available memory locations for adding information about new file additions.

In the definition of an "item", the "function" portion has not yet been accounted for. The function may be stored in memory locations next to the word or words representing the leaf or it may be "pointed to" by the third field of the leaf word (see Fig. 4-3).

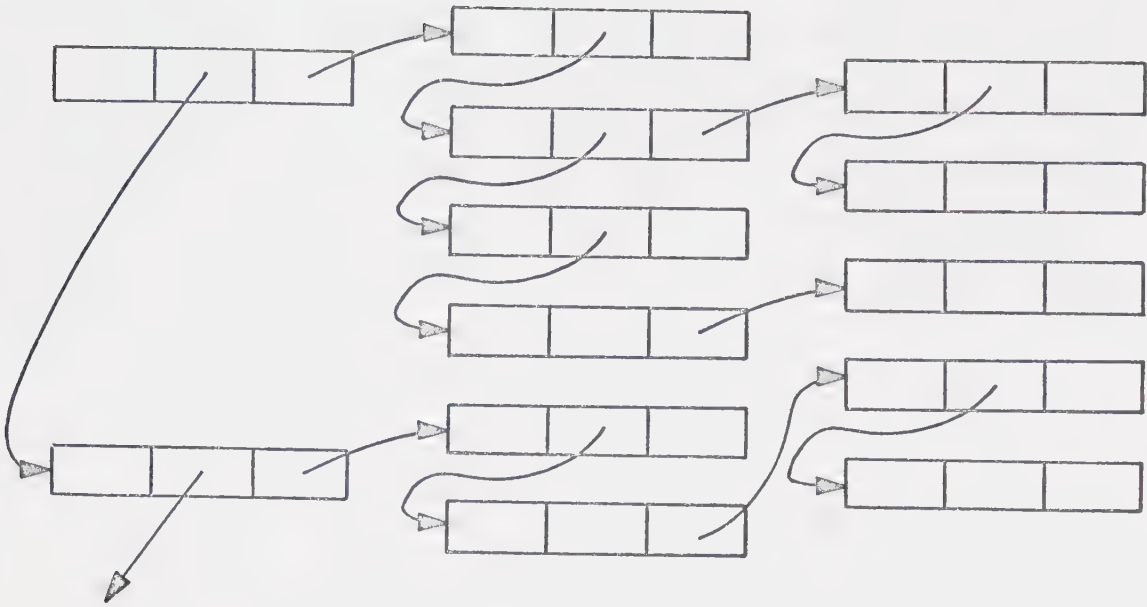


Fig. 4-3: Computer Representation of a Doubly-Chained Tree

| | | | | | | | | | | | |
|---|---|----|---|----|---|----|----|----|---|---|----|
| 1 | B | - | 2 | 8 | E | - | 11 | 15 | G | - | 20 |
| 2 | A | 3 | 4 | 9 | K | 13 | 14 | 16 | * | - | - |
| 3 | E | - | 5 | 10 | D | 17 | 18 | 17 | E | - | 19 |
| 4 | N | 6 | 7 | 11 | N | - | 12 | 18 | * | - | - |
| 5 | * | 8 | - | 12 | * | - | - | 19 | * | - | - |
| 6 | R | - | 9 | 13 | N | 15 | 16 | 20 | E | - | 21 |
| 7 | * | 10 | - | 14 | * | - | - | 21 | * | - | - |

Fig. 4-4: Memory Map for Tree of Fig. 4-2

Sussenguth has derived a formula for the expected search time. He has assumed that all terminal nodes are on the same level. The resulting formula is dependent on the number of nodes in a filial set (denoted by "s"). He has graphed search time versus filial set size and found that the minimum search time occurs when there are 3.6 nodes per filial set. He concludes that when "s" is at its optimum value the number of comparisons required is: $2.3 \log_{3.6} N = 1.24 \log_2 N$ where N is the number of items in the file. Thus, for the optimum case the expected search time is 24 per cent slower than the time required for a binary search.

References [13, 52] are concerned with Sussenguth's techniques either from the viewpoint of expansion of ideas or from the viewpoint of analysis of characteristics.

G. Salton [58] in his SMART system has adopted Sussenguth's techniques for dictionary manipulation. The SMART system has been in operation since 1964. SMART is essentially a document retrieval system. The system processes search requests against existing data bases and retrieves documents which closely correspond to the search queries.

One section of Salton's system involves thesaurus operations. In processing queries the system uses a computer-produced thesaurus to increase the correspondence between search request entries and entries in the data bases. Salton has adopted Sussenguth's double-chained method for storing thesaurus data for subsequent retrieval of thesaurus information or updating of existing thesaurus information.

The computer representation of the way in which Salton keeps track of the thesaurus data is the same as that mentioned previously (see Fig. 4-3). The thesaurus data is linked with dictionary entries which consist of semantic codes (concept numbers) and syntactic codes corresponding to a word stem in the data base being considered. The syntactic codes are used to combine word stems and suffixes into complete words in order to carry out a syntactic analysis. The tree entries are linked to dictionary entries (semantic and syntactic codes) via a pointer contained in the third field of the computer word associated with the final entry (character *) for the word being considered.

4.3 Method of Application of File Handling Techniques in THESAURI

The three file handling techniques mentioned were adopted to keep track of thesaurus entries and associated relationships in this application.

The binary search technique was adopted to keep track of the thesaurus entries. The constant accessing of terms was the main reason for choice of the binary search technique.

In addition, ideas advanced by Sussenguth were used in handling relationship information. However, rather than accommodating as many levels as are desired only two levels are necessary in accounting for relationship information. The first level entries contain information specifying the type of relationship, the location of more relationship information, and the location of information concerning terms related to the main term being considered by the designated relationship. The second level entries contain infor-

mation about the terms related to the main term being considered by the relationship mentioned above; they also contain information about the location of more information regarding terms bearing the same relationship to the main entry being considered. The details of the implementation of the modified Sussenguth method will be precisely specified later in the Chapter.

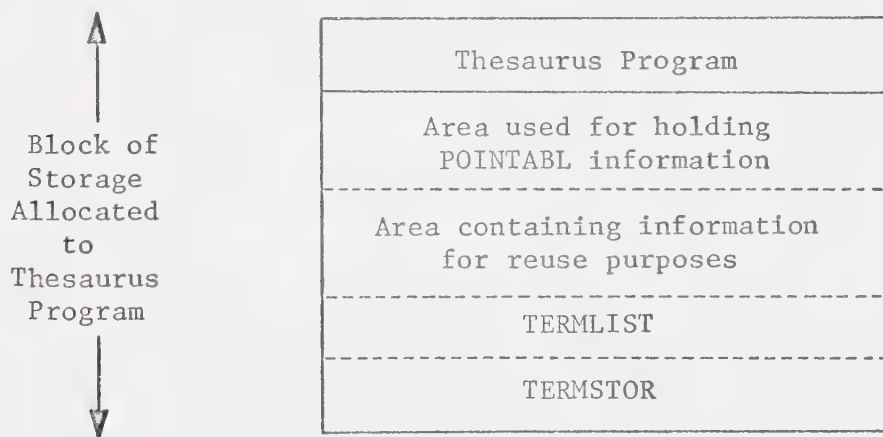
Chaining is used to associate relationship information (ie. first level entries) for a particular main entry and also to associate information concerning terms related to a particular entry by the same relationship. Chaining was chosen because it was felt that implementation of a more complex procedure on the small amount of information that would be involved, in any one case, would not result in any large increased savings in time.

The details of application of these three techniques will be made evident by the discussion following.

The key to the thesaurus program is the file structure associated with the handling of relationships for a given thesaurus entry. The file structure employed efficiently handles accounting for thesaurus entries and their relationships. The main feature of the program is the fact that with very minor changes a different thesaurus with totally different relationships can be handled. Thus one of the general requirements (number 3) stated in Section 4.1 has been satisfied.

The thesaurus program makes use of a large storage area which, in turn, is made up of four storage areas. The first area is used to hold information concerning thesaurus relationships (POINTABL infor-

mation) during program operation. The second area is used for purposes of accounting so information accepted by the program can be reused at a later time. The last two areas (TERMLIST and TERMSTOR) contain information concerning thesaurus entries. To make up the large storage area the program obtains space from the system via the MTS routine GETSPACE. The lengths of the storage areas TERMLIST and TERMSTOR are arbitrarily assigned values by the program. As will be seen in the explanation of the routines present in the program, these lengths can be varied by movement of the area TERMSTOR and, furthermore, the size of the large storage area can be altered. The appearance to the user of the thesaurus program and the partitioned storage block in computer memory is illustrated below.



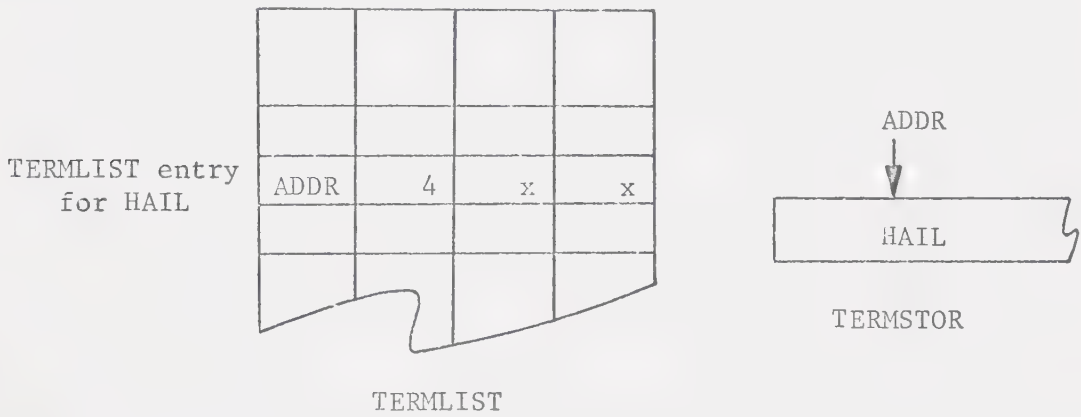
The storage area TERMLIST contains a collating sequence listing of terms or entries that make up the thesaurus the program is being used to manipulate. However, rather than contain the actual entries the four two byte fields comprising one TERMLIST entry contain information about thesaurus entries. Actual thesaurus entries are stored in the storage area TERMSTOR.

The appearance of the storage area TERMLIST is illustrated

below. Each position occupies two bytes of computer memory.

| | | | |
|---|---|---|---|
| | | | |
| A | B | C | D |
| | | | |
| | | | |
| | | | |

Field A contains the address of an entry in TERMSTOR and field B contains the length of this entry. Thus if the term being considered is HAIL, TERMLIST and TERMSTOR will appear as follows.



Field C contains the address, in the disk file containing POINTABL information for thesaurus entries, of the record containing POINTABL information for the entry in question, or -1 if the term does not have any subordinate relationships. Field D is used for the purpose of linkage to classification codes. The functions of this field will be explained in more detail at a later point.

A binary search is used on TERMLIST to obtain fast access to thesaurus entries and their associated relationships. When a thesaurus entry is accessed, for display purposes, for example, the thesaurus entries in TERMSTOR are compared against the entry about

which information is desired. The pointers in TERMLIST are first accessed to obtain the positions of the entries in TERMSTOR to be used for comparisons. The binary search technique was used here in order to be able to ascertain readily whether or not a particular term is already in TERMSTOR; use of the chaining method here would require serial searching which would be very time-consuming on a large file.

For each thesaurus entry with relationships a disk record (or records) exists which contains POINTABL information used for relationship accounting. Thus, if a thesaurus entry has relationships there is an associated disk record containing POINTABL information. When relationship information for a thesaurus entry is desired the disk record (or records) containing this information is read into the storage area mentioned previously. Certain techniques described in Section 4.2.5 concerning the "tree allocation method" are modified to fit this situation and are used in accounting for relationships between thesaurus entries.

Fig. 4-5 illustrates the appearance of disk records containing POINTABL information used for relationship accounting.

The appearance of POINTABL information for a thesaurus entry is illustrated below.

| | | | |
|---|---|---|---|
| | | | |
| A | B | C | D |
| | | | |
| | | | |
| | | | |

First Record Containing Relationship Information for Thesaurus Entry

| | | | | | |
|--|---|--|---------|---|---|
| 0 | 2 | 4 | 6 | 8 | 10 |
| Record Number of Record Containing More Information | AVAIL Available Location For More Information | Initial Relationship Address | FLINDEX | Available Location Due to Deletion | Available Location Due to Deletion |
| 12 | 14 | 16 | 256 | | |
| Available Location Due to Deletion | Available Location Due to Deletion | POINTABL Information for Thesaurus Entry | | | |

Second Record Containing Relationship Information for Thesaurus Entry

| | |
|--|-----|
| 0 | 256 |
| POINTABL Information for Thesaurus Entry | |

Fig. 4-5: Format of Records Used in Relationship Accounting

Each POINTABL entry is made up of four two byte fields. For each term in the thesaurus for which there exists subordinate relationships there is information concerning these relationships stored as POINTABL information. POINTABL entries can be of two different types; these two different types will be explained in the discussion following.

In regard to the tree allocation method the nodes on the first level (ie. roots) contain information about the relationships belonging to the main term. The four fields for each entry corresponding to a first level node can be explained as follows:

- A. forward pointer -- This field contains the POINTABL address of more information concerning first level nodes (ie. more relationships belonging to the main entry). If there exist no more first level nodes containing further relationship information this field contains -1.
- B. backward pointer -- This field contains the address in POINTABL of information concerning another first level node preceding the one we are now considering. If there are no preceding first level nodes this field contains -1. This field is used to keep relationships for a main thesaurus entry in a particular order for printing purposes.
- C. key -- This field contains a code indicating the relationship between the main term and the terms about which information is contained in the POINTABL entry under consideration. These codes correspond in meaning to print table entries. Thus, the meanings of the codes can be changed by altering print table

entries. At present the codes and their meanings are as follows:

| | |
|----------------|--------------------|
| 1 -- main term | 4 -- broader term |
| 2 -- use | 5 -- narrower term |
| 3 -- used for | 6 -- related term |

- D. filial set pointer -- This field contains the address in POINTABL of information concerning the first thesaurus term which is related to the main term by the relationship designated by the key.

The filial sets, which are accessed initially by the filial set pointer, are also represented by pointers in POINTABL. The four fields for each entry corresponding to a second level node (filial set) can be explained as follows:

- A. forward pointer -- This field contains the POINTABL address of another filial set containing information about a term related to the main thesaurus entry by the same relationship as the term referred to by the filial set presently being considered. If there are no other filial sets for that relationship for the main thesaurus entry being considered this field contains -1.
- B. backward pointer -- This field contains the POINTABL address of information concerning another thesaurus entry which was entered into the thesaurus at a time prior to the time that the entry being considered was entered. If there are no preceding second level nodes this field contains the POINTABL address of the first level node.
- C. address -- This field contains the address in TERMSTOR of the term whose relationship to the main term is specified by the key

of the first level node that this particular filial set is associated with.

D. length -- This field contains the length of the term in TERMSTOR whose TERMSTOR address is given by the field "address".

Thus an eight byte POINTABL entry can contain two kinds of relationship information: (1) information concerning the type of relationship; and (2) information concerning the thesaurus entries related to the main thesaurus entry by the relationship indicated by the information contained in (1).

Fig. 4-6 gives a view of: (1) the two storage areas TERMLIST and TERMSTOR; (2) disk records containing POINTABL information for thesaurus entries; and (3) the relationship between (1) and (2).

Explanation of a simple example should make evident the relationship between the storage areas and disk records containing relationship information. In Fig. 4-7 the contents of the various storage areas are displayed.

In TERMSTOR the terms HAIL and ICE begin in locations 900 and 904, respectively. In TERMLIST information concerning the two terms (address and length) is entered in the first two columns. In the third column the 64 entered in the row containing information about the term HAIL is the disk address in the file of the record containing relationship information for the term. The 160 in the row containing information about the term ICE indicates that 160 is the disk address in the file of the record containing relationship information for the term. The numbers 150 for the term HAIL and 170 for the term ICE are numbers used in the indexing and searching

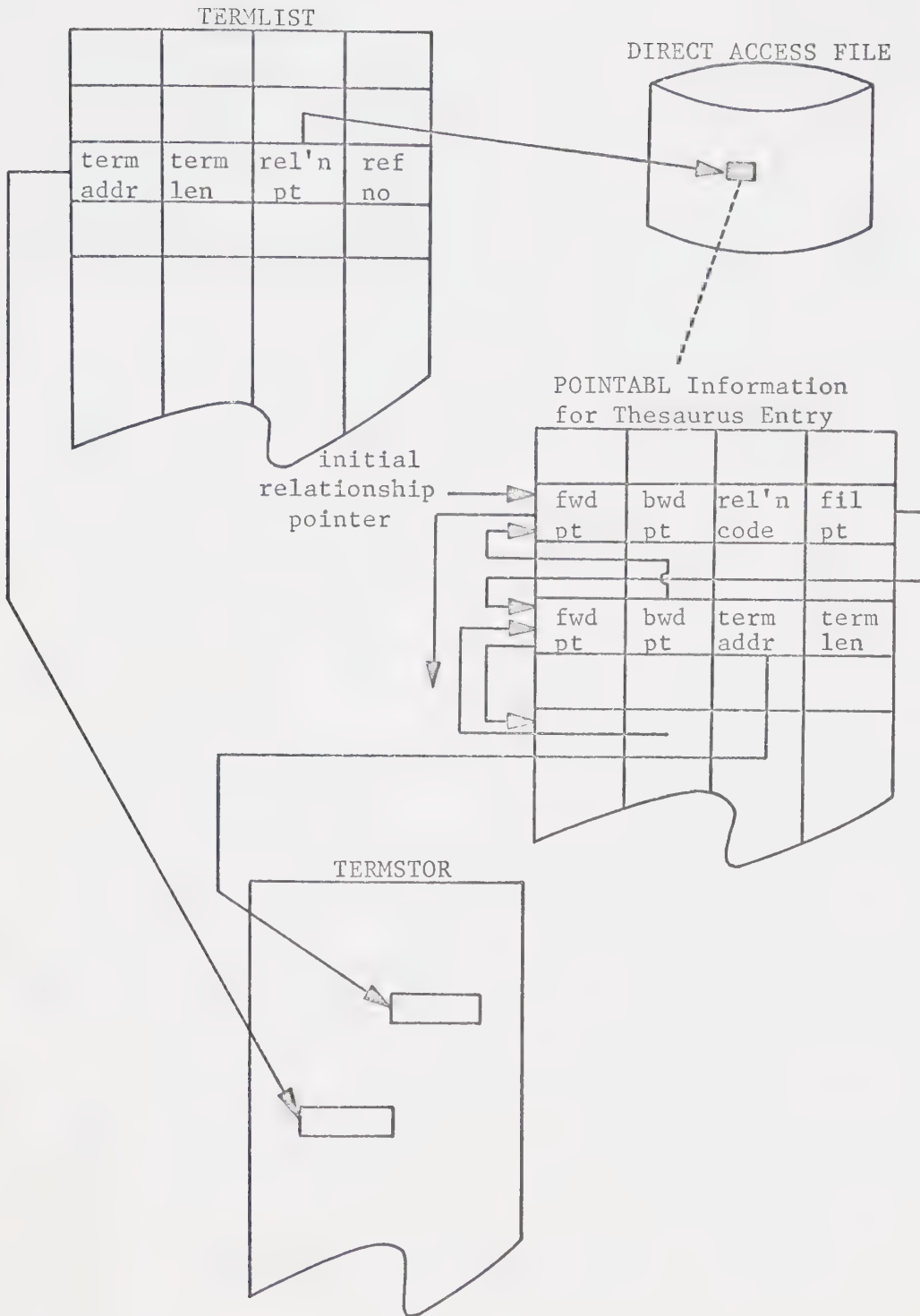


Fig. 4-6: Diagrammatic Representation of Information Accounting

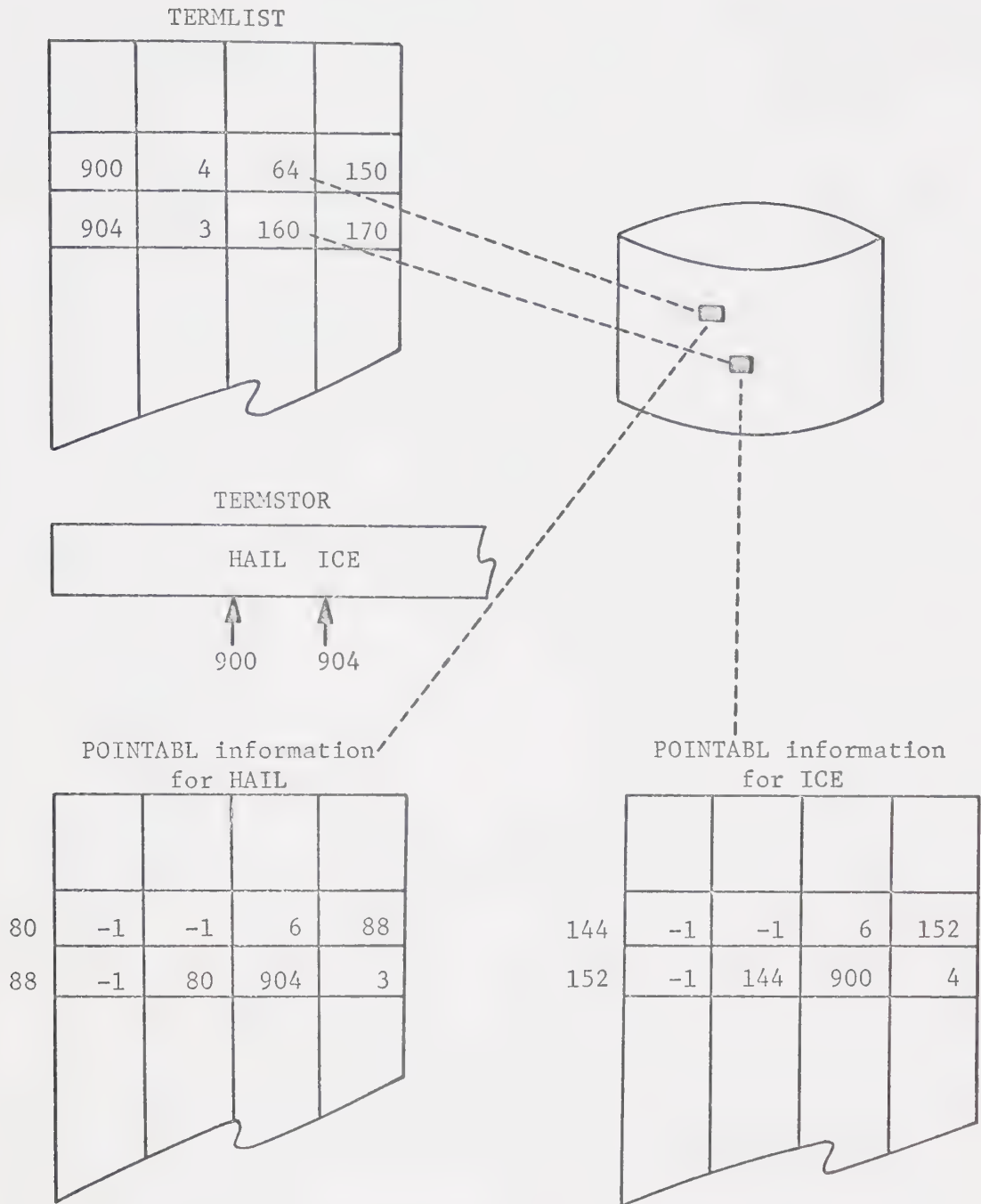


Fig. 4-7: Example of Information Accounting

phases of the system. These numbers are the record numbers in an MTS line file where UDC class numbers corresponding to thesaurus entries are located or should be located.

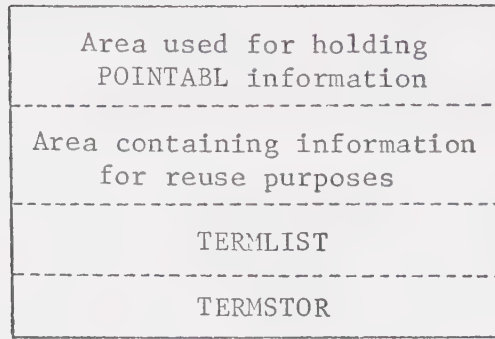
In location 80, in POINTABL information for the term HAIL, the -1 in the forward pointer position indicates there are no more relationships. The -1 in the backward pointer position indicates that this is the first relationship for the term. In the third column the 6 indicates that the relationship is a related term. The 88 in the fourth column gives the POINTABL address of the first filial set for this relationship. Starting in location 88 the -1 in the forward pointer position indicates that this is the only filial set for the given relationship. The 80 in the backward pointer position refers back to the row containing the first level node information. The 904 in the third column is the TERMSTOR address of the term ICE while the 3 in the fourth column indicates the length of the term. These pointers have set up the following relationship:

```
HAIL
  RT  ICE.
```

The information contained in locations 144 and 152 in POINTABL information for the term ICE handle the reciprocal relationship. This is the following:

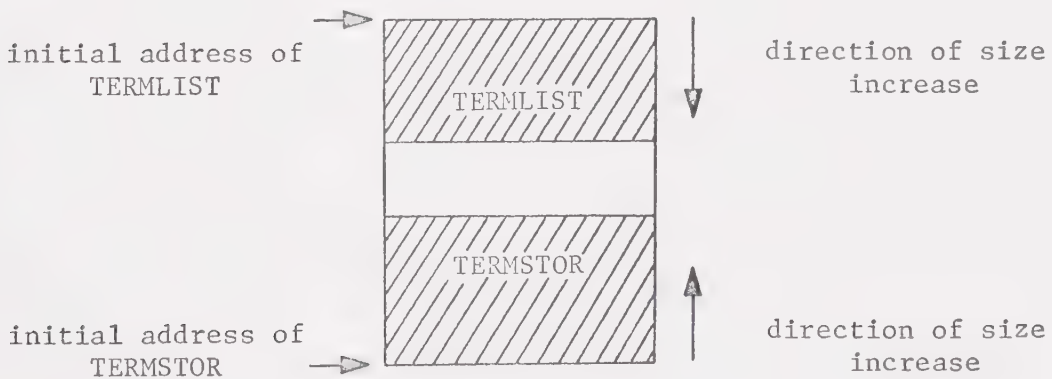
```
ICE
  RT  HAIL.
```

As was previously mentioned, the large storage area, obtained from the system for use by THESAURI, to the user, appears as follows:



The routine MOVETABL, which will be more fully described in the section describing the routines in THESAURI, carries out movements of the storage area TERMSTOR in 256-byte pieces in order to make use of the space allocated to the program due to alteration of the "space desired" parameter used in calling the MTS system routine GETSPACE.

The way in which the tables increase in size is illustrated below.



In order to minimize the number of movements required due to entries in TERMLIST overlapping or threatening overlapping onto entries in TERMSTOR, entries in TERMSTOR are made from the back to the front.

4.4 Description of Module THESAURI

4.4.1 Introduction

The thesaurus program for this thesis is written in 360 Assembler Language. Assembler was chosen for a variety of reasons. The main reason was the need for storage conservation. Another reason was that the algorithms written readily lent themselves to manipulations via assembler language instructions.

The program was written in modular form to facilitate easy modification if desired. The modular form also made errors reasonably easy to detect. In this section a description of the program is given.

4.4.2 Thesaurus Entry and Copy

The reusability of thesaurus entries and relationship information established through use of THESAURI is necessary. By reusability is meant the storage of a copy of the results of one terminal session or batch run being available, unaltered, for use during a session or run conducted at a later time or date. In this area THESAURI exhibits a lack of system independence. However, implementation of the module under time-sharing systems other than MTS would necessitate only application of system idiosyncrasies to the framework already present in this phase of the module.

THESAURI exists as a load module in an MTS line file, as do the other modules in the system. When operation of the programming system is initiated the modules in the system are loaded into core for subsequent use.

The module THESAURI is called by MONITOR. Three parameters are passed to THESAURI by MONITOR. These parameters are: (1) a code which indicates that either a new thesaurus is to be constructed (value N) or that the existing thesaurus file is to be used during the terminal session or batch run (value 0); (2) the accessible command table length which allows a user to access all or only some of the commands in the command table depending on whether or not the correct signon code was entered by the user at signon time; (3) the runcode indicating the mode in which the system is being utilized. The possible values are B for a batch run and T for a terminal session.

The runcode is necessary because of the fact that when the MTS system subroutine SCARDS reads from MASTER SOURCE the length read is returned to the calling routine. When MASTER SOURCE is the card reader the length returned is 80 whether or not 80 characters are punched on the card. When MASTER SOURCE is an on-line terminal the length of the character string entered by the user is returned. For this reason it is necessary to have a short routine in THESAURI which computes the length of a character string read from a punched card. This routine is activated during batch runs. A sequence of two blanks must be found before the end of the character string is recognized.

THESAURI occupies approximately 4K of storage. Rather than have a large storage area defined at the end of the code comprising the program it was felt that obtaining space from the system was a better approach. The MTS system subroutine GETSPACE is used to

obtain space from the system. One of the parameters GETSPACE requires is the number of bytes desired. By modifying this value the amount of space allocated to a thesaurus can be modified. The theoretical before and after appearance of THESAURI is illustrated below.



After GETSPACE has been called to obtain memory for the storage of information necessary for maintenance of a thesaurus the MTS system subroutine READ is called to read in the MTS sequential file which contains the required information. The READ subroutine is called using the sequential option. The READ subroutine is called as many times as is needed to read in the sequential file containing thesaurus information. This phase is carried out even if the "construct new thesaurus option" is given. This is done to obtain information needed for a procedure which will be explained shortly. Fig. 4-8 shows the contents of the MTS sequential file containing thesaurus information.

At this point the MTS system subroutine GDINFO is called in order to obtain the file and device usage block (FDUB) pointer. Obtaining this value is necessary for modification of the READ, WRITE and LAST pointers for the sequential file containing the thesaurus information before it is rewritten at the end of a terminal session or batch run.

The code indicating establishment of a new thesaurus or usage

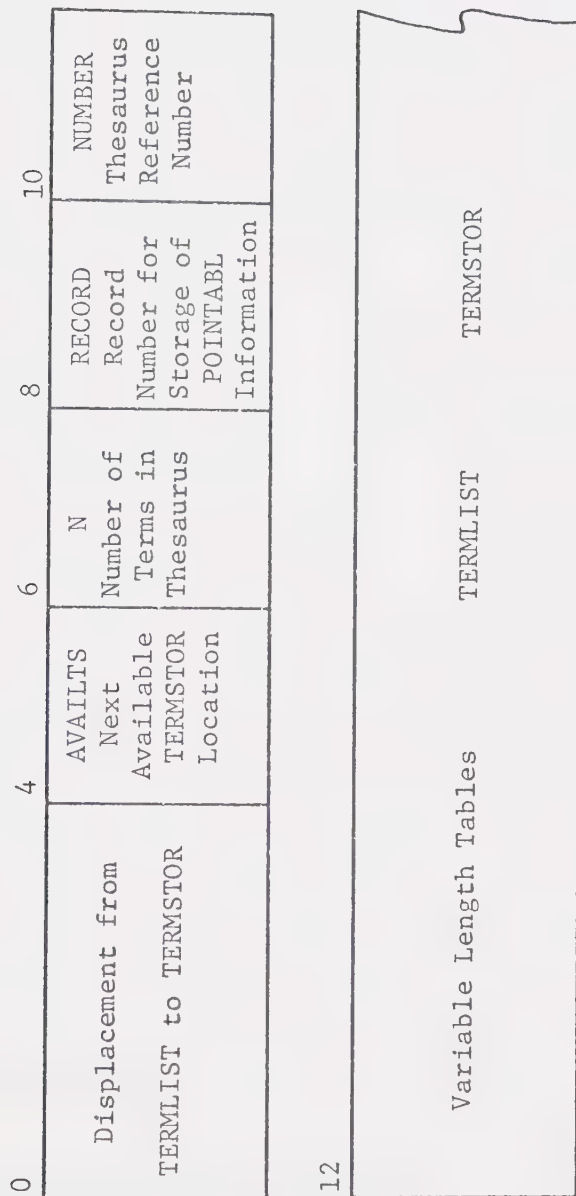


Fig. 4-8: Format of Sequential File Containing Thesaurus Information

of existing thesaurus information is now tested. If construction of a new thesaurus is desired a branch to the routine which initializes storage contents is taken; otherwise the routine which reads commands and transfers control to the various routines within the program is entered.

When a user enters the command "exit" the module THESAURI must write out information necessary for continued access to the thesaurus information established during the terminal session or batch run. In order to accomplish this the MTS subroutine POINT is called to reinitialize the pointers used in reading and writing the sequential file. The constants necessary for future use of the thesaurus are stored in the beginning locations of the output area. The MTS system subroutine WRITE is then called as many times as necessary in order to write out the thesaurus information. The MTS system subroutine FREESPAC is then called to release the storage that was obtained with GETSPACE. Finally control is transferred to the module MONITOR.

The above described procedure is followed each time control is transferred to THESAURI. As was mentioned at the beginning of this section the reusability feature is necessary. The framework described here should be followed in any implementation under another time-sharing system.

4.4.3 Command Recognition Routine

The routine COMMAND is responsible for transferring control to the various routines in the program depending on user entered

commands. COMMAND does a sequential scan of the entries in the command table comparing the user entered command against command table entries until either a match is found or no more commands in the table can be used for comparison purposes. The latter mentioned case applies when one of the two conditions below are met: (1) the user, because of the signon code entered when initially signing on, is allowed access to only some of the commands in the command table (At the present time if the user enters the wrong signon code he has access only to the commands "display" and "exit" in the module THESAURI.); (2) there are no more commands in the command table. As explained in Section 4.4.2 describing the procedure adhered to in entering THESAURI and exiting from THESAURI, the module MONITOR sets the accession code after recognizing the user entered signon code. The accessible command table length is simply a parameter passed from the module MONITOR.

COMMAND transfers control to the appropriate routine in the program depending on the user entered command. The format in which users must enter commands is given in the Appendix and described in Section 5.4. All routines return to the routine COMMAND after completion. The routine directly accessible via COMMAND are PRINT, RITEFILE (exit routine), NEWTERM, KILLTERM, SPELLING, CHECSPAC, MOVETABL, and INITIAL.

4.4.4 Initialization for Construction of New Thesaurus

The routine INITIAL handles the initialization of parameters and addresses to facilitate construction of a new thesaurus. The

addresses initialized are: (1) the address of the storage area TERMLIST; (2) the address of the storage area TERMSTOR.

The values of constants used by routines within the program are set to zero. The mask on a branch instruction within the routine BINSRCH is modified so that the first new thesaurus entry is entered without the checking required for the subsequent entering of entries.

The initialization of storage area addresses to certain values does not mean that these addresses are fixed for the remainder of the life of the thesaurus being manipulated by the program. Use of the routine MOVETABL can cause the starting address of TERMSTOR to be changed.

INITIAL can be called by a user entering the correct command or automatically if, upon entry to THESAURI, a new thesaurus copy is requested.

4.4.5 Checking of Space

The routine CHECSPAC, when initiated by the appropriate command, displays to the user two values: (1) the number of terms in TERMLIST; (2) the amount of space, in bytes, remaining between TERMLIST and TERMSTOR.

The amount of space remaining between TERMLIST and TERMSTOR is calculated by the following formula:

$$\begin{aligned} \text{space remaining} &= \text{address of TERMSTOR} - \text{address of TERMLIST} \\ &- (\text{number of terms in TERMLIST} \times 8) + \text{next available TERMSTOR} \\ &\quad \text{location.} \end{aligned}$$

This routine should be initiated by the user at fairly fre-

quent intervals to ascertain the status of the thesaurus he is manipulating via THESAURI. THESAURI does not perform checking to determine whether or not there is enough space remaining in a storage area for further additions. When the space remaining in a storage area has been exhausted MOVETABL can be called to reposition the storage area TERMSTOR or, alternatively, the space parameter in THESAURI can be modified, the program reassembled, and then MOVETABL called to reposition the storage area TERMSTOR.

4.4.6 Binary Search Routine on Term Information

The binary search routine BINSRCH handles establishment of the location of and insertion of term information in TERMLIST. This routine is used by all routines in THESAURI that access term information in a nonsequential manner.

BINSRCH can operate on TERMLIST in two different ways:

(1) search for terms and if they are not accounted for in TERMLIST add information concerning them to TERMLIST; (2) search for terms only.

The actions performed by BINSRCH depend upon parameters initialized prior to its being called. The parameters that BINSRCH expects to receive are: (1) the length of the term being considered; (2) the initial address of the storage area TERMLIST; (3) the initial address of the storage area TERMSTOR; (4) the number of terms accounted for in TERMLIST; (5) an indicator specifying the option desired.

If a term to be added to the file is not accounted for in TERMLIST the routine COREMOVE is called to accomplish space creation for the new term information in the correct position in TERMLIST.

Regarding the workings of BINSRCH there is one point that is worth special mention. Within BINSRCH the "termcode" (relationship code) of the entry being considered is tested. If its value is 1 indicating a main entry, the TERMLIST address for that entry is stored in "tlpcodel". If "termcode" has a value other than 1 the TERMLIST address of the entry is stored in "tlpother". This process is carried out in order to have the addresses in TERMLIST of information concerning the main thesaurus entry ("tlpcodel") and the relationship entry being considered ("tlpother") available for use in NEWTERM, the routine which handles the addition of terms and relationships. These addresses are used in the initialization of storage for the calling of various routines which perform necessary operations in NEWTERM. Because there is a possibility that relationship entry information might be added to TERMLIST in a position that results in main term information being repositioned, it is necessary to test the main term against the incoming relationship term in order to ascertain whether or not modification of "tlpcodel" is required.

4.4.7 Movement of Storage Areas

The routine COREMOVE handles the moving of storage tables or positions of storage tables either to reposition a table for arrangement purposes or to move a portion of a storage table to make room for new entries or to delete existing entries.

COREMOVE is on a general enough basis that it can be made to operate on any storage area. Four parameters must be passed to COREMOVE: (1) the address indicating where the core movement is to begin;

(2) the length to be moved ie. the amount of core to be moved; (3) the number of positions that the storage area is to be moved; (4) an indicator specifying the direction of movement ie. up or down. The routines within THESAURI that call COREMOVE initialize these parameters before passing control.

The routine uses "256-byte moves" in carrying out the storage movements. This means that 256-byte sections of a table or storage area are moved at one time. Naturally the last move made by COREMOVE involves an area of length 256 bytes or less. The number 256 stands out here because the 360-Assembler Language MVC instruction allows for up to 256 bytes to be moved at one time.

COREMOVE operates on two storage areas (TERMLIST, TERMSTOR) in the program. Movements are made on TERMLIST either to make space available for new term information or to delete existing term information. Thus, movements of TERMLIST might involve all of TERMLIST or only a portion of TERMLIST. Operations on the storage area TERMSTOR are performed on the complete table. Repositioning of this table is carried out when the space allocated to the program (ie. the space obtained from the system by GETSPACE) is altered. These conditions, however, make no difference to COREMOVE. In all cases the four parameters described previously must be initialized.

In the program COREMOVE is called by the routines KILLTERM, SPELLING, MOVETABL, and BINSRCH.

4.4.8 Addition of Terms and Relationships to the Thesaurus

The routine NEWTERM handles the addition of terms and relationships to a thesaurus. This routine is used in adding new terms and relationships to a thesaurus or in establishing relationships between existing thesaurus entries.

NEWTERM initially reads the user specified main thesaurus entry. BINSRCH is called to add term information to TERMLIST if the entry is not already accounted for or to obtain the location of information in TERMLIST concerning the entry. After this has been done a process of reading the relationship entries and their relationship codes, and establishing the relationships and any reciprocal relationships takes place.

As can be seen in the command syntax specifications (see Appendix and Section 5.4) a relationship entry consists of a term followed by a numeric code which indicates the relationship of the term to the already specified main term. At present THESAURI is operative with the codes and relationships specified in Table 4-1. Also specified in Table 4-1 are the meanings of the codes and the associated reciprocal codes and meanings if any exist.

| <u>Code</u> | <u>Meaning</u> | <u>Reciprocal</u> <u>Code</u> | <u>Meaning</u> |
|-------------|----------------|----------------------------------|----------------|
| 1 | main entry | 0 | |
| 2 | use | 3 | used for |
| 3 | used for | 2 | use |
| 4 | broader term | 5 | narrower term |
| 5 | narrower term | 4 | broader term |
| 6 | related term | 6 | related term |

Table 4-1: Codes and Meanings

The allowed for relationships correspond to the relationships present in the U.S. Department of the Interior Water Resources Thesaurus [84]. The meanings of the relationship codes are controlled by a print table used by the routine DISPLAY in displaying thesaurus entries and their associated relationships. It should be noted that these meanings and allowable relationships are by no means fixed. By altering the codes, reciprocal relationship codes, and print table entries, totally different relationships and reciprocal relationships can be handled by THESAURI. Thus, general requirement 3 stated in Section 4.1 has been satisfied. However, reassembly of the program is required.

After a user enters a relationship entry and relationship code the following events take place: (1) if the relationship entry is to be used in indexing and query formulation, a "reference number" is assigned to the entry via the routine REFNUMB only when a number has not yet been assigned to the entry; (2) the routine INITRERO, which initializes storage for call to the routine which adds relationship information to POINTABL for the thesaurus entry, is called; (3) the routine RELAROUT, which adds relationship information to POINTABL in the disk record (or records) containing relationship information for the thesaurus entry in question, is called; (4) the routine CLEANUP is called; this routine restores the unused POINTABL addresses initially obtained from FREELIST back into FREELIST in the disk record containing relationship information for the thesaurus entry in question; (5) the initial relationship address for the main thesaurus entry is stored in the disk record containing relationship

information for the entry; (6) the disk record containing relationship information is written into the file containing relationship information for the thesaurus entries.

Initially these events take place with the main entry and the relationship entry being considered in their rightful status. After this has been accomplished the reciprocal relationship code for the specified relationship code is obtained. If it is not zero in value the positions of the main thesaurus entry and the relationship entry are reversed; that is, the relationship entry is thought of as being the main entry and the main entry is thought of as being the relationship entry, related to the pseudo-main entry by the relationship designated by the reciprocal relationship code. The steps outlined above are then followed to establish the reciprocal relationship.

After the relationship and reciprocal relationship, if any, have been established, more relationship entries and codes can be entered by the user. This process stops when the user enters any one character or no characters.

The transference of consideration of main entry to relationship entry and relationship entry to main entry, and the initialization required in order to utilize some of the routines in the above mentioned six steps, should make the reader aware of the necessity of the two parameters "tlpcodel" and "tlpothor" mentioned in the description of the routine BINSRCH.

4.4.9 Routine to Print File or Portion Thereof

The routine PRINT handles the printing of thesaurus entries and relationships. Three options are available to the user. The options allow for: (1) the printing of the terms making up the thesaurus without associated relationships in collating sequence order; (2) the printing of the terms making up the thesaurus including relationships, thesaurus reference numbers, and associated classification codes if they exist in collating sequence order; (3) the printing of one thesaurus entry with relationships and thesaurus reference number if they exist. The third option is most likely to be used when the system is operational in on-line mode. Options one and two should be used only when the system is operational in batch mode. A printed copy of the complete thesaurus with relationships, reference numbers and classification codes is equivalent to the bound copy of the English Electric Thesaurofacet [3] which was discussed earlier.

PRINT is centered on a contained routine called DISPLAY. DISPLAY handles the printing of thesaurus entries, and, possibly, relationships, reference numbers, and associated classification codes, depending on the setting of various exits with the routine. The disk record (or records) containing relationship information is read into the area used in holding POINTABL information for a thesaurus entry. The relationship information for an entry is accessed as if it were "in core". To print associated classification codes the MTS file, containing the classification codes corresponding to thesaurus entries, is accessed using the thesaurus reference number

as the record number. The record accessed contains the classification codes which correspond to the thesaurus entry that the reference number is assigned to. This will be explained in greater detail in the section dealing with thesaurus and classification code linkage.

DISPLAY uses an execute instruction (EX) to move terms and relationship entries to the print line for subsequent printing. The codes indicating the relationships of the relationship entries to main thesaurus entries are used to access a print table containing abbreviations for the meanings of the relationships. The correspondence between print table entries and relationship codes for the version of THESAURI now being used is given in Table 4-2.

| <u>Relationship</u> <u>Code</u> | <u>Print Table</u> <u>Entry</u> |
|------------------------------------|------------------------------------|
| 2 | USE |
| 3 | UF |
| 4 | BT |
| 5 | NT |
| 6 | RT |

Table 4-2: Print Table Entries

As was mentioned in the explanation of the routine NEWTERM the relationship code meanings can be changed by altering corresponding print table entries.

Three short routines within PRINT supervise the operation of DISPLAY. This supervision includes setting of exits within DISPLAY depending on desired options and the calling of DISPLAY. The three short routines are called ALFALIST (initiated when a collating sequence list of terms is desired), COMPLIST (initiated when a com-

plete list of terms, relationships, reference numbers, and classification codes is desired), and PONETERM (initiated when printing of one thesaurus entry with relationships and reference number, if they exist, is desired).

PONETERM uses BINSRCH to locate the user specified term in TERMLIST in order to obtain information relevant to subsequent printing. ALFALIST and COMPLIST go through TERMLIST sequentially in order to obtain the thesaurus entries to be printed.

4.4.10 Routine to Reposition Storage Area TERMSTOR

The routine MOVETABL handles the movement of the storage area TERMSTOR. The user specifies: (1) the direction of movement (up or down); (2) the number of positions the storage table is to be moved. Usually MOVETABL is used to reposition TERMSTOR for one of the following reasons: (1) there may be no room in a storage area for additional information; (2) more computer storage has been allocated to THESAURI (ie. the space parameter has been modified) and a downward movement of TERMSTOR is desirable in order to make use of this added storage; (3) less computer storage has been allocated to THESAURI (apply condition to (2) above).

This routine initially checks the command sequence for validity and, if valid, sets up the necessary parameters, depending on the user entered command, for the subsequent calling of COREMOVE. Naturally the address of the repositioned storage table is modified to take the movement into account.

4.4.11 Initialization for Call to Routine which Establishes Relationships

The routine INITRERO initializes registers and storage for subsequent calling of RELAROUT which adds relationship information to POINTABL for a thesaurus entry. Basically INITRERO: (1) reads in the disk record (or records) containing relationship information for a thesaurus entry if it exists or initializes a disk record if the entry in question has no relationships associated with it; (2) sets up the two available POINTABL addresses required by RELAROUT; (3) sets the initial relationship pointer required by RELAROUT either from a value in the disk record (see Fig. 4-5), if the entry already has relationship information associated with it, or from the first available POINTABL address if relationships have not yet been established for the thesaurus entry; (4) sets mask values in conditional branch statements within RELAROUT; and (5) converts the code for the relationship into a binary value.

The POINTABL addresses set by INITRERO are available from two sources. Initially FREELIST for the entry in question is accessed to ascertain whether or not any POINTABL locations are available due to deletions. These addresses are used if any are available. The fact that the second available address may or may not be used makes it necessary to modify the contents of the eight byte POINTABL entry referred to by this address. If the second address is not used the routine NEWTERM, which supervises the addition of term and relationship information to a thesaurus, calls FREELADD which stores this address in FREELIST. On subsequent use of this address the trick

mentioned above allows the address to be correctly considered by INITRERO. The FREELIST addition routine FREELADD is also possibly called by INITRERO. This is done if POINTABL entries, available for use because of previous deletions, allow for more than two entries under any specified address in FREELIST.

If there are no POINTABL addresses in FREELIST available for use, the next available POINTABL location pointer AVAIL is used in setting up one or both of the required addresses.

The actions performed by INITRERO in setting up the two POINTABL addresses required by RELAROUT are further explained in Section 4.4.15, which deals with garbage collection.

4.4.12 Addition of Relationship Information to POINTABL

The routine RELAROUT handles the addition of relationship information to POINTABL for a thesaurus entry. The actions performed by RELAROUT are independent of the type of relationship being considered. RELAROUT inserts information into POINTABL storage locations adhering to the requirements described for relationship accounting. RELAROUT expects certain parameters to be passed to it. The routine INITRERO described previously initializes storage for subsequent calling of RELAROUT. The actions RELAROUT performs are as follows: (1) if this is the only relationship entry of its kind (ie. no other relationship entries appear under this relationship code for the thesaurus entry being considered) in POINTABL for the thesaurus entry being considered: (a) relationship information is set in POINTABL ie. forward and backward pointers for the relationship are set, the

relationship code is set, and the filial set pointer is assigned a value, and (b) the term information is set in POINTABL ie. forward and backward pointers for the filial set are set, and the length and address of the relationship entry are set from values passed to RELAROUT; (2) if there already exists relationship entries for the relationship code being considered part (b) in step (1) above is the procedure carried out. Under point (1)(b) above it is important to note that relationships for a particular thesaurus entry are in increasing order according to the thesaurus relationship code which is specified when relationships between thesaurus entries are established. This is done mainly for printing purposes. This requirement necessitates RELAROUT's modification, if required, of the forward and backward pointers for relationship information. This has somewhat complicated the workings of RELAROUT.

This routine expects two available POINTABL addresses to be passed to it; one of these addresses is always used for the storage of information while the other may be used. The origins of these two addresses are described in the descriptions of garbage collection for POINTABL locations and the initialization of storage by INITRERO for subsequent calling of RELAROUT.

On return to the calling routine RELAROUT returns the initial relationship address in register 5. It is possible that this address may have been modified by RELAROUT so restoring of the address in the disk record containing relationship information for the term being considered must be carried out upon return from RELAROUT. The reason for its possible modification is the "ordering by relationship code

requirement" noted in the second paragraph above.

4.4.13 Routine to Change Spelling of Thesaurus Entries

The routine SPELLING handles the changing of term spellings. The routine uses the READ routine two times; the first "read" obtains the term whose spelling it is desired to change and the second "read" obtains the new spelling of the term. The term whose spelling is to be changed must have the following characteristics or an error message will be returned to the user: (1) it must be accounted for in TERMLIST; and (2) it must have relationships associated with it. The new spelling of the term must have one of the following properties: (1) it must not be accounted for in TERMLIST; or (2) if it is accounted for in TERMLIST it must not have relationships associated with it. In instances where these requirements cannot be met the routines DELETE and NEWTERM can probably be used to accomplish the desired end. Here the user is faced with the responsibility of term deletion and assignment of relationships to the new term. Regarding the second restriction on the term whose spelling is to be changed; this was instituted because it was felt that a term accounted for in TERMLIST with no attached relationships could be just as easily deleted (using the routine DELETE) and then the new spelling could be added (using NEWTERM).

SPELLING first reads both of the required terms and establishes their validity or lack of it according to the above-mentioned requirements. This sequence was adopted in order to eliminate the possibility of having eliminated the original spelling from TERMLIST

and then finding that the new spelling was invalid.

The routine COREMOVE is then called to eliminate information concerning the original term spelling from TERMLIST. After this has been accomplished the relationship information for the new term (relationship information is the same as for the old spelling) is accessed. The spelling of the term is changed from the old to the new in any of the relationships for which reciprocal relationships exist.

4.4.14 Routine to Delete Terms and/or Relationships

The routine KILLTERM handles the deletion of a term and all of its relationships or the deletion of individual relationships of a term. Reciprocal relationships involving the deleted terms or relationship entries are automatically deleted.

In the case of deleting a term and all of its relationships the routine COREMOVE is used to delete information concerning the term from TERMLIST.

KILLTERM must be capable of handling the deletion of individual relationships for thesaurus entries and also the deletion of thesaurus entries and all of their relationships. For this reason a routine called DELETREL located within KILLTERM handles: (1) the alteration of POINTABL entry values needed because of deletions; (2) the obtaining of the reciprocal relationship code for the relationship entry being deleted; (3) the alteration of POINTABL entry values for the reciprocal relationship entry if a reciprocal relationship exists; (4) the rewriting of disk records containing relationship information for thesaurus entries. It is in this routine

that additions are made to FREELIST, the storage area containing available POINTABL addresses for each thesaurus entry for which relationships exist. The POINTABL locations made available for use because of the deletion of relationship information for thesaurus entries are added to FREELIST.

The supervisory section of the routine KILLTERM deciphers the command sequence read from the terminal or card reader. Initialization is made for subsequent calling of DELETREL, the functions of which have already been described. If a term and all of its relationships are to be deleted the portion of DELETREL which performs manipulations necessary for the deletion of reciprocal relationship entries, if they exist, is accessed first. In the case of deletion of an individual relationship, DELETREL is entered from the beginning; the specified relationship is deleted if it exists and any reciprocal relationship is also deleted if it exists. In the case of deleting a term and all of its relationships, term information is deleted from TERMLIST, and POINTABL locations containing relationship information are made available for subsequent additions to POINTABL. For this reason it is unnecessary to enter DELETREL from the beginning and individually delete information concerning relationships. Disk records in which information is modified and which may be used at a later time are written out.

Examinations of the workings of the routines KILLTERM and SPELLING reveal that, for the file structure used in tabulating relationship information, handling deletions and spelling changes is somewhat awkward but the advantages attained in generality more

than offset this awkwardness.

4.4.15 Garbage Collection

When terms are deleted from a thesaurus or relationships are deleted from existing thesaurus entries space becomes available in the storage area POINTABL in the disk records containing relationship information. This is, of course, assuming that the term deleted has relationships associated with it or that the thesaurus entry has relationships associated with it so that the deletion of individual relationships is possible. The space that becomes available consists of eight byte entries from four possible sources. The four possible sources are: (1) the eight byte entry containing relationship information (ie. relationship code, filial set pointer, etc.) for the thesaurus entry deleted or the thesaurus entry from which relationships are deleted; (2) the eight byte entry containing information concerning the term which is related to the thesaurus entry in the manner designated by the relationship code mentioned in (1); (3) the eight byte entry containing reciprocal relationship information (ie. relationship code, filial set pointer, etc.) for the reciprocal relationship entry; (4) the eight byte entry containing information concerning the main thesaurus entry which is related to the thesaurus entry in the manner designated by the relationship code mentioned in (3).

Consider the case when a relationship entry for a thesaurus entry is deleted. In this instance the space mentioned in (2) above becomes available. If the deleted relationship entry is the only entry related to the main thesaurus entry in the manner designated

by the code mentioned in (1) the space mentioned in (1) becomes available. When the relationship between entries was initially established reciprocal entries were made if reciprocal relationships existed. If reciprocal relationships existed the space mentioned in (4) above becomes available. Furthermore if there is only one relationship entry for this particular relationship the space mentioned in (3) also becomes available.

The available space can be used in accounting for more relationship information. For this reason when new relationship information for a particular thesaurus entry is entered in the storage area POINTABL, the available POINTABL locations for the entry in question, that have been set in the routine that deletes terms and all relationships or individual relationship entries, are used first. When the areas available have been exhausted the next available POINTABL location pointer AVAIL for the entry in question is used as an available address for the insertion of information.

The storage area where the addresses available due to deletions are stored is named FREELIST (see Fig. 4-5). At present enough space in each disk record for relationship information is allocated in FREELIST to allow for 4 addresses. This reasonably small number was chosen because it was felt that entering terms and relationships would occur more frequently than deleting; subsequently any addresses available because of deletions would be readily used. The index FLINDEX is used to indicate the availability of entries in FREELIST and also the availability of space

in FREELIST for further additions. Both FREELIST and FLINDEX must be stored in the disk record containing relationship information for the thesaurus entry. Fig. 4-5 shows the position of FREELIST and FLINDEX in the disk record.

As mentioned previously the routine (KILLTERM), which deletes terms and relationships or individual relationships, makes entries in the area FREELIST after deleting entries and relationship or relationships alone. The computer code accomplishing this task also makes modifications to the contents of these available POINTABL locations. This is done to trick the routine (INITRERO) which initializes storage for the call to the routine (RELAROUT) which adds relationship information to POINTABL locations in the disk record containing relationship information for a thesaurus entry. For any one call to RELAROUT one eight byte entry is always used and another eight byte entry may be used. Thus two addresses in POINTABL must be passed to RELAROUT; one is always used and the other may be used. These two addresses are set first from FREELIST entries if there are any, and secondly from AVAIL (the next available POINTABL location for the thesaurus entry in question). The possibility of use of the second address makes it necessary to check to see if it has been used on return from the routine RELAROUT for one of the following reasons: (1) if the second has been used and AVAIL was used for designating available POINTABL locations AVAIL requires modification; (2) if FREELIST addresses were used as the available POINTABL locations and the second designated address was not used by RELAROUT an entry must be made in FREELIST to specify this second address as

an available POINTABL location. A routine called CLEANUP handles the above-mentioned tasks. This routine is part of the routine which supervises the addition of entries and relationship entries to a thesaurus (NEWTERM).

4.5 Conclusions

In Section 4.1 the requirements for the on-line thesaurus program were specified. As has been shown THESAURI adequately fulfills these requirements. The improvement over the previous University of Alberta programs should also be noted. In the previous programs all of the information required for a terminal session or batch run is in core. Paging plays a large role in the cost of program operation. Furthermore, the method of implementation limits the size of the thesaurus that can be handled by the programs. In THESAURI the method of implementation necessitates that some of the information is in core and that some of the information is stored on disk. Thus, the cost of disk accesses plays a major role in the cost of program operation. However, the size of the program is smaller and paging costs are not as large as those incurred in the previous programs. Increased cost of program operation, if any, is more than offset by the increased capabilities of THESAURI.

CHAPTER V

THE THESAURUS CENTERED SYSTEM

5.1 Introduction

As stated in Chapter 1 a thesaurus linked to a classification scheme is a powerful tool which, in the hands of a librarian or classifier, can be used for classification purposes. By accessing the thesaurus with indexing terms deemed applicable to the article or book being classified, additional indexing terms may be suggested by the thesaurus relationships. Furthermore, the linkage with a classification scheme will give classification codes to be used in the identification of the article or book. A similar procedure may be followed to obtain the class code, which may be used to locate a previously classified article or book.

A scheme of this type is even more powerful when operations are carried out by a computer, especially with an efficient computer program handling thesaurus manipulations at the head of the system. The application of a scheme of this sort, with storage and retrieval of pertinent information carried out by a computer in an on-line environment, is very attractive from a user's point of view. The mass of paper necessary in any manual implementation is eliminated for the most part; the associated tedious lookup is also eliminated. To retrieve wanted information a user need only sit at a computer terminal and type in simple commands.

From the above we can see that the thesaurus classification link carried out in a man-machine type environment in itself is an

improvement over a completely manual thesaurus and classification link. Even more powerful is the linkage of the thesaurus classification structure with an information storage and retrieval system. An application of this type allows for: (1) indexing and classifying information in a subject area covered by a thesaurus and computer storage of the resulting data; (2) searching of data generated by the classification and indexing of documents in a certain subject area based on the thesaurus classification linkage; (3) linking with a complete bibliographic data base, the records of which might contain title, authors, keywords, classification codes, and accession numbers.

The first two of these ideas are explored in some detail in a further extension of the research for this thesis. The following are needed to implement these two capabilities: (1) a means of linking a thesaurus and a classification scheme in a manner which is efficient and allows reasonably easy machine manipulation; (2) a means for making additions to data bases or changes to data base entries; (3) a means for searching the data bases mentioned in (2).

Such a combination will provide a tool that efficiently serves many purposes; certain of these are discussed below.

5.2 Thesaurus and Classification Scheme Linkage

The thesaurus and classification linkage is a problem in machine implementation. Should the classification codes associated with a thesaurus entry be handled in the computer as part of the

thesaurus or should they be handled separately? It is true that for a complete printed listing of a thesaurus the associated classification codes must appear in the printout. However, a tie in with the module THESAURI to ensure such a printout on every demand would further complicate the file structure. From a computer point of view handling the codes separately is much more efficient. If this is done a way of accessing codes for printing purposes and other manipulations is required. In the implementation described below the classification codes were handled as a separate part of the system.

The method used to link thesaurus entries with classification codes is quite straight forward and reasonably efficient. In TERMLIST a two byte entry containing a record number is attached to thesaurus entries which are used in indexing and searching. The numbers are assigned, by the module THESAURI, sequentially, from zero, as new entries used in indexing and searching are entered into the thesaurus. The numbers serve as record numbers in an MTS file where classification codes, associated with the thesaurus entries used in indexing and searching, may be located. A module in the system allows a user to access records in the file containing classification codes for purposes of making new entries in the file, retrieving entries, or updating entries. This module, called INDEXING, is explained in more detail in Section 5.3.3. Thus, a user, after making an addition to the thesaurus, can use INDEXING and the record number that has been assigned automatically by THESAURI to store the classification codes associated

with the thesaurus entry in the classification codes file. The record numbers also serve as indexing entries in the data base containing records consisting of document accession numbers and thesaurus reference numbers used in indexing the documents.

To illustrate the above, Fig. 5-1 gives the relationship between the thesaurus entries used in indexing and searching, the classification codes file, the data base containing records consisting of document accession numbers and thesaurus reference numbers, and the data base containing records consisting of document accession numbers and keywords used to index the documents.

As explained previously, initiating the routine PRINT (with the complete list option) in the module THESAURI results in a complete listing of thesaurus entries, relationships, reference numbers and classification codes. In this instance the file containing classification codes is accessed using the thesaurus reference numbers as record numbers. A sample print-out of a thesaurus entry using the "complete list" option is given below.

| HAIL | THESAURUS REFERENCE NUMBER | CLASSIFICATION CODES |
|------|----------------------------|----------------------|
| UF | ICE STONES | |
| BT | ICE | |
| | PRECIPITATION | |
| | FREEZING | |
| RT | GRAUPEL | |
| | STORMS | |

From above the thesaurus reference number for the indexing and searching term HAIL would appear as an indexing entry for any information indexed by the term HAIL in the file consisting of document

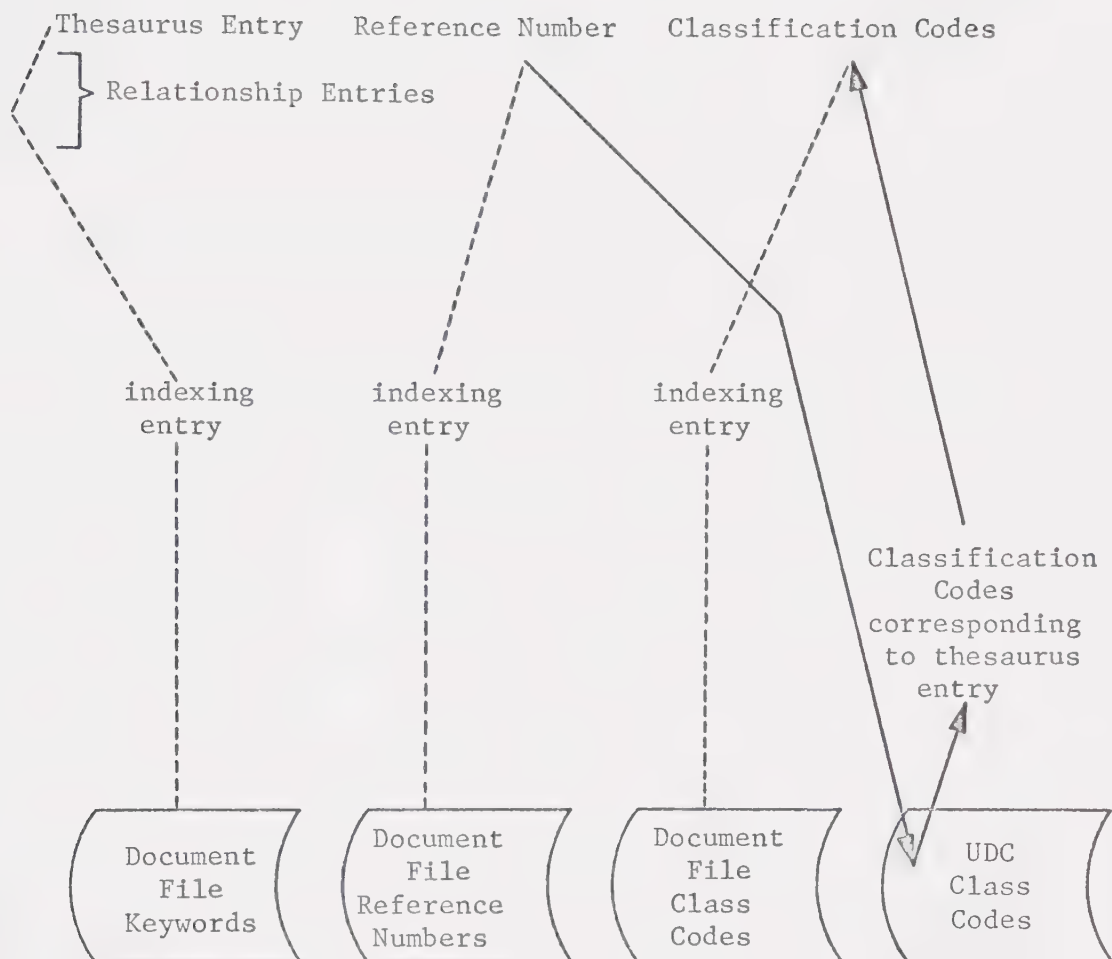


Fig. 5-1: Diagrammatic Representation of Relationship Between Term Information and Files in System

numbers and thesaurus reference numbers. The classification codes are from a record in the file containing classification codes. The thesaurus reference number serves as the record number in the file where the classification codes corresponding to the term HAIL are located.

5.3 Overall System Configuration

5.3.1 Introduction

The requirements for a specific thesaurus-based classification linked information storage and retrieval system have been stated in Section 5.1. The module (THESAURI) which handles thesaurus manipulations was described in Chapter 4. Four other modules were written to handle the additional operations. Assembler was chosen as the programming language for reasons of brevity of actual programs and speed of operation. The system is modular in nature for the following reasons: (1) reasonably easy modification is facilitated; (2) possible expansion of the system can be accomplished with a minimum number of problems; (3) implementation in stages was allowed for by separating functions in the system; (4) a modular approach prevents the overall problem from becoming too complex to implement efficiently; (5) a modular approach allows dependence on the time-sharing system under which the system is operative to be confined, for the most part, to one module.

The overall configuration for the system is given in Fig. 5-2. The files associated with the system as well as the

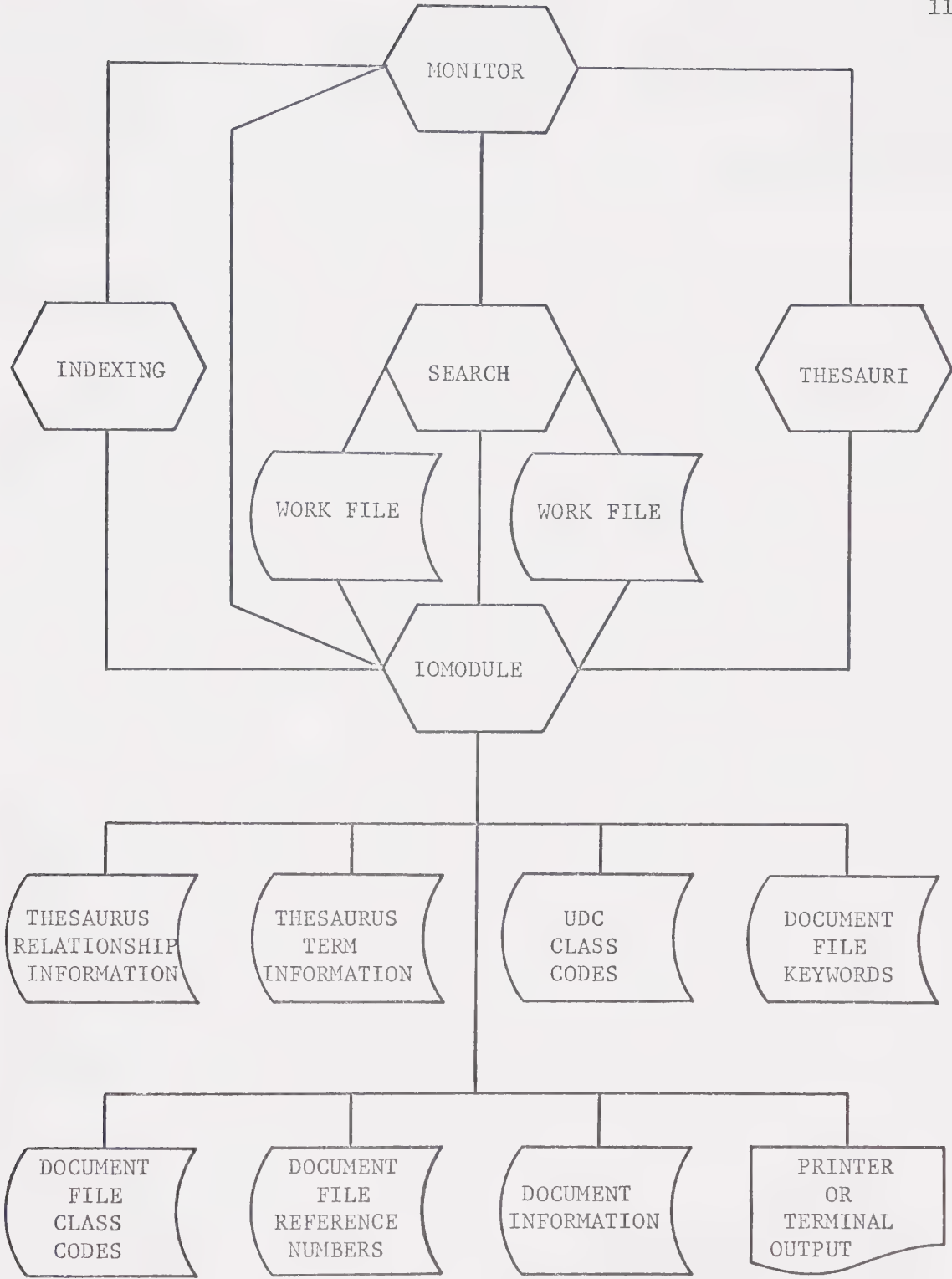


Fig. 5-2: Overall Configuration for System

program modules involved are included. A general diagram indicating system capabilities is given in Fig. 5-3.

Descriptions of these four modules follow. The modules are MONITOR, INDEXING, IOMODULE, and SEARCH.

5.3.2 Module MONITOR

The module MONITOR, as its name suggests, "monitors" the system. The other modules in the system (ie. INDEXING, SEARCH, and THESAURI) are activated by MONITOR in either batch or on-line modes by commands entered by the user. The module MONITOR passes necessary parameters to these "called routines". When the programming system is initially activated the user is presented with the query "SIGNON CODE?". At this point the user should respond with a sequence of characters which result from an arithmetic operation on the date. If the user is correct in his reply he is allowed access to all commands available. If his reply is incorrect the user is allowed only to access information in the files associated with the system. He is not allowed to make changes to any of this information. This facility is built into the system to prevent an unauthorized user from destroying information in the system files.

5.3.3 Module INDEXING

The module INDEXING allows a user to access the following files: (1) the file containing classification codes corresponding to thesaurus entries; (2) the file consisting of records containing document accession numbers and keywords used in indexing the documents; (3) the file consisting of records containing document

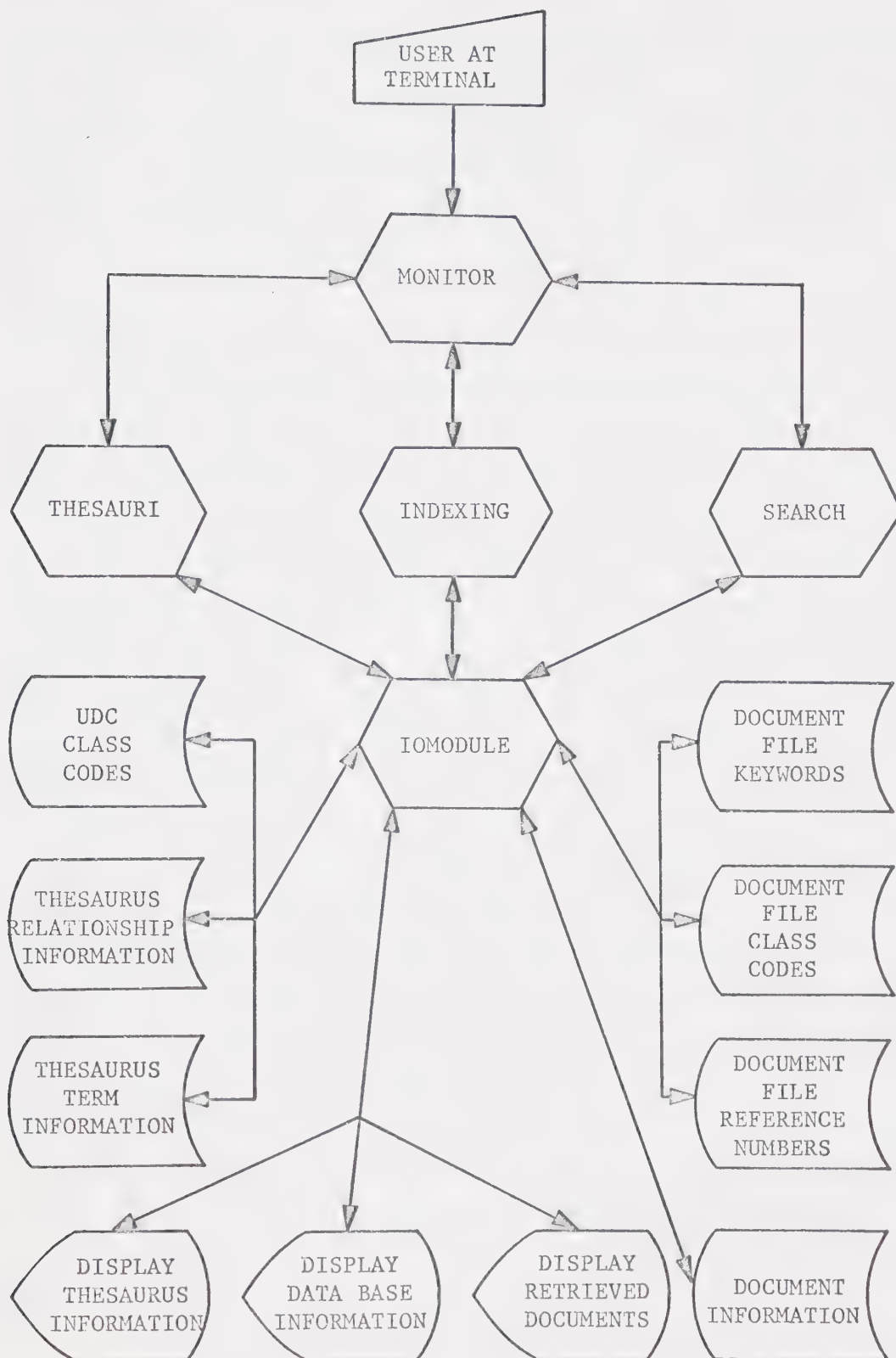


Fig. 5-3: Diagrammatic Representation of System Capabilities

accession numbers and classification codes used in indexing the documents; (4) the file consisting of records containing document accession numbers and thesaurus reference numbers corresponding to keywords; (5) the file consisting of records containing document information. The formats for the records in these files are given in Fig. 5-4.

By using the module INDEXING a user can retrieve any record from the above specified files, update any record in the above specified files, or enter new records into any of the above specified files. For example, if a user wanted to find the classification codes associated with the thesaurus entry HAIL, he would first obtain the thesaurus reference number for HAIL. He would then use the thesaurus reference number and the correct command sequence (assuming that the module INDEXING has control) to obtain the classification codes for the entry HAIL.

By a simple procedure (MTS sequence of commands) operations can be performed on these files in a manner which is divorced from the programming system. For purposes of initial creation or subsequent updating on a large scale this might be the best procedure to adopt. However, in using the system in an interactive environment the module INDEXING is necessary. Its necessity should become obvious when the reader is aware of the functions of all of the modules in the system.

In utilizing the module INDEXING the user must enter the commands in the format described in the command language description. The module examines the command sequence for validity. If

| | | | | | | |
|-------------------------------|-----------------------|-------------------------------|-----------------------|-------------------------------|-----------------------|--|
| UDC Classification Code | D E L I M | UDC Classification Code | D E L I M | UDC Classification Code | D E L I M | |
|-------------------------------|-----------------------|-------------------------------|-----------------------|-------------------------------|-----------------------|--|

UDC Classification Code File

| | | | | | | | | | | |
|---------------------------------|-----------------------|---------|-----------------------|---------|-----------------------|---------|-----------------------|---------|-----------------------|--|
| Document Accession Number | D E L I M | Keyword | D E L I M | Keyword | D E L I M | Keyword | D E L I M | Keyword | D E L I M | |
|---------------------------------|-----------------------|---------|-----------------------|---------|-----------------------|---------|-----------------------|---------|-----------------------|--|

Keyworded Document File

| | | | | | | |
|---------------------------------|-----------------------|-------------------------------|-----------------------|-------------------------------|-----------------------|--|
| Document Accession Number | D E L I M | UDC Classification Code | D E L I M | UDC Classification Code | D E L I M | |
|---------------------------------|-----------------------|-------------------------------|-----------------------|-------------------------------|-----------------------|--|

Classed Document File

| | | | | | | | | |
|---------------------------------|-----------------------|----------------------------------|-----------------------|----------------------------------|-----------------------|----------------------------------|-----------------------|--|
| Document Accession Number | D E L I M | Thesaurus Reference Number | D E L I M | Thesaurus Reference Number | D E L I M | Thesaurus Reference Number | D E L I M | |
|---------------------------------|-----------------------|----------------------------------|-----------------------|----------------------------------|-----------------------|----------------------------------|-----------------------|--|

Numbered Document File

Fig. 5-4: Record Formats

valid INDEXING calls the module IOMODULE after setting up the parameter lists as required.

5.3.4 Module IOMODULE

The module IOMODULE handles all the input output operations in the system. All other modules in the system call this module. As mentioned previously, one of the reasons for the modular nature of the system was, if possible, to isolate system dependence. This module is MTS specific in that the input output operations it handles are specific to the MTS time-sharing system presently in operation at the University of Alberta. If the programming system were implemented on another time-sharing system this module would have to be rewritten. Its characteristics, after rewriting, would adhere to the specifics necessary to carry out input output operations under the time-sharing system.

Calling modules pass IOMODULE five parameters. The five parameters are: (1) the address of a key indicating the I/O operation to be performed; (2) the address of the area which information is to be read into or written from; (3) the address of a length field where the length read or to be written is stored; (4) the address of the record number in the MTS file that is to be accessed or operated on; (5) the address of the logical I/O unit that the MTS file is assigned to.

The module IOMODULE returns to the calling module a return code in register 15. The return code is that set by the MTS input output routines referenced by the module.

The module IOMODULE calls the MTS system subroutines READ, WRITE, SCARDS, and SPRINT. READ and WRITE access user specified files while SCARDS reads from the MTS MASTER SOURCE and SPRINT writes onto the MTS MASTER SINK. In batch mode the MASTER SOURCE is the card reader and the MASTER SINK is the line printer. In on-line mode the MASTER SOURCE and MASTER SINK are the terminal.

5.3.5 Module SEARCH

5.3.5.1 Introduction

The module SEARCH allows for weighted term searches on three of the data bases which presently exist in the programming system: (1) the data base containing document accession numbers and keywords representing document content; (2) the data base containing document accession numbers and U.D.C. classification codes representing document content; (3) the data base containing document accession numbers and thesaurus reference numbers corresponding to keywords which represent document content. It should be noted that these three data bases can be generated by making use of the thesaurus classification scheme linkage. A further discussion of other methods of data base generation is found in Section 6.3. The keywords and thesaurus reference numbers are related directly to the thesaurus entries; the classification codes correspond to the thesaurus entries and would be related to a classification system.

Fig. 5-4 gives the formats of the records in the above specified data bases. The logic allowed is OR nested within AND and NOT.

When using the module SEARCH the user is initially presented with the query "DATA BASE?". The user can make three legitimate responses: (1) "classed" -- if he wants to search the data base containing classification codes for documents; (2) "keyword" -- if he wants to search the data base containing keywords representing document content; (3) "refnumb" -- if he wants to search the data base containing reference numbers corresponding to keywords which represent document content.

After establishing which data base is to be accessed the module displays the command "ENTER QUESTIONS". The module expects question input (according to the command specifications) until it encounters the user response "end". Thus the search is made with respect to a batch of questions. At this point the module accesses the desired data base attempting to match table entries whose values are dependent on question input against data base entries. "Hits" for questions along with the input questions are written into a file as they are obtained. After all records in the data base being considered have been tested for correspondence against question input the MTS system utility SORT is called as a subroutine (by the module SEARCH). This process sorts the question input and any "hits" obtained into an order such that specific questions and "hits" for these questions appear in the output one after another. A routine in the module then handles the printing of this sorted file presenting the user with output. Sample output can be viewed in Section 5.6.5. The basic techniques used in this module were designed by L.H. Thiel and H.S. Heaps [81]. The techniques used

allow reasonably fast searching and only require accessing of records in the data base once for any one batch of questions.

5.3.5.2 Internal Workings of the Module SEARCH

The module SEARCH prepares tables from the question input and when there is no more question input compares table entries and data base entries for correspondence.

A system flowchart for the procedures carried out in the module SEARCH is given in Fig. 5-5.

The module allows for 10 questions to be batched together for any one run. This number can be expanded if desired. The module creates three tables: (1) QUEPARTB -- question parameter table; (2) POSTABLE -- position table; (3) TERMTABL -- term table.

TERMTABL is made up of 16 byte entries. Fourteen of these sixteen bytes are for actual terms, one byte for the length of the term, and one byte for a pointer to a four byte entry in the table POSTABLE. Throughout the module one byte is used for length and pointer storage because one byte can represent the numbers 0 through 255.

The one byte pointer in TERMTABL entries points to a four byte entry in POSTABLE. POSTABLE entries consist of four one byte entries: (1) QUES -- one byte whose value indicates in which question the term from TERMTABL occurs; (2) PARAM -- one byte indicating which parameter this term is, in the question; (3) WEIGHT -- one byte indicating the weight of the term in the question; (4) NEXT-ONE -- one byte pointer to another entry in POSTABLE containing

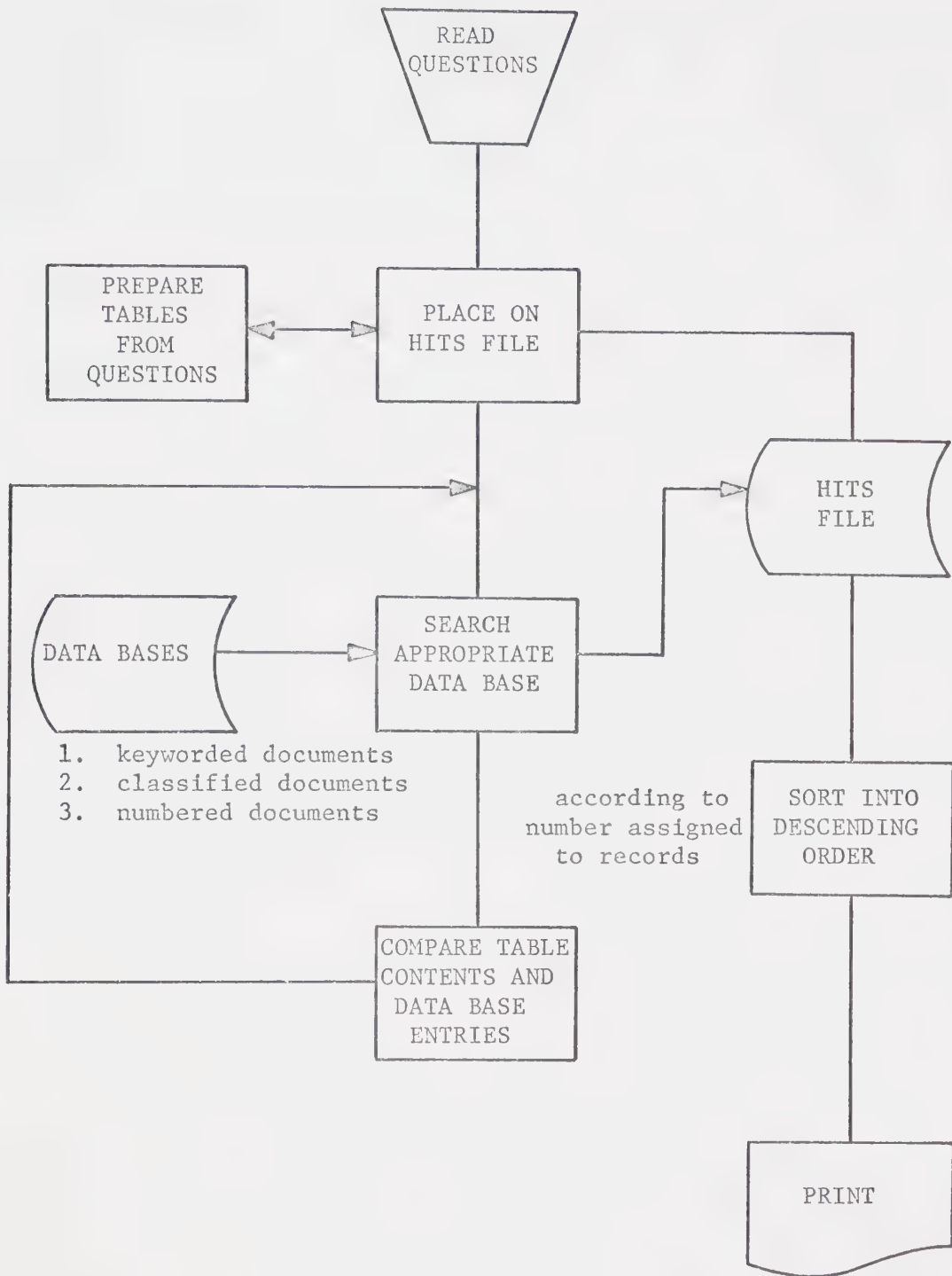


Fig. 5-5: System Flowchart for Search Procedure

information about the occurrence of this same term in other questions.

The table QUEPARTB also consists of four byte entries. Each four byte QUEPARTB entry is made up of four one byte entries. The number of the question in the table QUEPARTB is also the number of the question in the batch of questions prepared for input to the module. The four one byte entries in each QUEPARTB entry are:

(1) QUEPAR -- one byte whose value is set by the comparison of TERMTABL and POSTABLE entries and data base entries; (2) REQPAR -- one byte whose value depends on question logic (explained in greater detail later); (3) THRESH -- one byte whose value must be met or exceeded in order for the data base record to qualify as a "hit" for the question; (4) SUMWT -- one byte whose value is set by the comparison of TERMTABL and QUEPARTB entries and data base entries. If the term in TERMTABL is an entry in a data base record the WEIGHT of the term in the question(s) it appears in is added to SUMWT for the question(s).

The phases in the module SEARCH can roughly be specified as follows:

(1) Reading of Questions and Construction of Tables

In this stage a binary search routine is used to enter question entries into TERMTABL. The entries are in ascending order according to the computer's collating sequence. The structure of an entry in TERMTABL is given in Fig. 5-6. Entries are also made in POSTABLE. The question number, the parameter number, and the weight of the term are set for each term encountered. If an entry appears more than once in a batch of questions the pointer NEXT-ONE is used to

indicate another position in POSTABLE where more question information concerning the entry can be found. In QUEPARTB the question threshold value is set (THRESH from the value specified on the PROFILE specification). Also set is the one byte field REQPAR. The contents of this field, after setting, specify which entries must be present in data base records for the record to qualify as a "hit" (AND logic), which entries must not be present (NOT logic), and which entries are optional (OR logic). If the logic specified for the question entry is AND the appropriate parameter bit is set to 1. If the logic specified is NOT the parameter bit is set to 0. All question records are written into a file along with a number for subsequent sorting.

(2) Comparison of Data Base Records and Created Tables

In this phase data base records for the data base being considered are read sequentially. For each entry in a data base record TERMTABL is accessed via a binary search to ascertain whether or not the entry is in TERMTABL. If the entry is not in TERMTABL the next entry in the record being considered is tested for its presence and so on. If the entry is in TERMTABL the weight of the term in the questions that the term appears in is added to SUMWT for the appropriate questions in QUEPARTB. The one byte field QUEPAR is also altered by setting the appropriate bit to 1 (depending on PARAM value for the particular question).

(3) Testing for Hits

In this phase the fields REQPAR and QUEPAR are tested for equivalence (done for each question for every data base record considered). If

they are not equal SUMWT and QUEPAR are set to zero and the fields for the next question are examined. If REQPAR and QUEPAR are equivalent SUMWT and THRESH are compared. If SUMWT is greater than or equal to THRESH in value the data base record is deemed a "hit" for the question being considered. The associated document accession number is written into the "hits" file along with a number computed for sorting purposes. SUMWT and QUEPAR are zeroed for the question and the next entry in QUEPARTB is examined. Phase 2 and phase 3 are continued until there are no more records to consider in the data base.

(4) Sort and Printing

In this phase the records written into the "hits" file are sorted into descending order using the computed number as the sort field. The method in which the numbers are computed results in questions and "hits" for the questions ending up in order (question appears first then "hits" for the question). The printing phase just prints the records from the sorted "hits" file onto the terminal or line-printer depending on whether the session was in on-line or batch modes.

Fig. 5-6 shows the relationship between TERMTABL, POSTABLE, and QUEPARTB.

After processing one set of questions the user is again queried for the data base on which he wishes to operate. He can reinitiate the procedure just described or enter "exit" and return to the module MONITOR.

The entries in the data base records for the data base

consisting of documents indexed by keywords must be truncated to a length of six bytes. In preparing questions for subsequent searching of this data base the user must specify terms that are six characters in length.

If the module is using the data base consisting of documents indexed by thesaurus reference numbers comparisons are done on four byte fields.

If the module is using the data base consisting of classified documents a slightly different situation arises. A user might want to search on a general concept, a location, or a time period. For this reason a length field was included in TERMTABL entries. When comparing entries in TERMTABL and data base entries the length used is that of the shorter entry.

The module SEARCH can be easily altered to handle other data bases if necessary.

5.4 Command Language

A command language has been developed for the system. The command language is a series of rules for denoting commands and statements recognizable by the programming system. Basically the command language description gives the format in which commands and statements must appear in order to be recognized by the system.

A metalanguage (a language used to describe other languages) is used to describe the command language. The special symbols used to describe the command language are a combination of the symbols used in Backus Normal Form (BNF) and in Inverted Notations. The

meanings of the special symbols used in defining the language and the command language specifications appear in the Appendix.

5.5 Command File

In order to minimize the amount of knowledge a user must have regarding MTS, the MTS commands needed to initiate operation of the system could be in a file called "commandfile". In any implementation under another time-sharing system a similar approach could be adopted dependent, of course, on system characteristics. For MTS, to initiate system operation the user need only enter:

```
$source commandfile
```

The file "commandfile" can be created using the following sequence:

```
$create commandfile
$empty commandfile
$set commandfile
$number 1,1
$$run monitor+indexing+search+thesaurus+iomodule
    scards=*msource* sprint=*msource*
    2=-sortinp 3=-sortout 4=filename1
    5=filename2 6=filename3 7=filename4
    8=filename5 9=filename6 10=filename7
$unnumber
```

where:

- (1) filename1 -- name of the MTS sequential file containing thesaurus information;
- (2) filename2 -- name of the line file containing thesaurus information regarding relationships;
- (3) filename3 -- name of the line file containing classification codes which correspond to thesaurus entries;
- (4) filename4 -- name of the line file containing records made up of

document accession numbers and keywords used in indexing the documents;

- (5) filename5 -- name of the line file containing records made up of document accession numbers and classification codes used in indexing the documents;
- (6) filename6 -- name of the line file containing records made up of document accession numbers and thesaurus reference numbers corresponding to thesaurus entries which are used to index the documents;
- (7) filename7 -- name of the line file containing records made up of document information (title, author, keywords, etc.).

It should be noted that through use of different files the programming system can be used to handle information storage and retrieval in more than one field. In information centers there is a need for general systems which can function with information from different disciplines. One of the major points considered in the design of this system was the need for such generality.

5.6 Sample Sessions

5.6.1 Introduction

The workings of the modules in the system can only be illustrated by a series of comprehensive terminal sessions. Samples of these sessions which illustrate most of the allowable features are given and explained below.

In all of the examples given the user entered commands are in lower case while the computer response is in upper case.

From the examples given the reader should be made aware of the man-machine interaction features of the system, the functions carried out by the different modules in the system, and the capabilities of an information storage and retrieval system employing a thesaurus approach linked with a classification scheme. As an aid to the reader in understanding the method of linkage between thesaurus and associated classification codes, listing of a term, relationships, and classification codes was given in Section 5.2.

5.6.2 MONITOR

As explained in Section 5.3.2 the module MONITOR passes control to modules in the programming system depending on user entered commands. The signon code entered by the user establishes the accessibility code for the system. The user, to have access to all of the commands, must sign on with the correct code. At present, the correct code consists of the numeric calendar date in month, day, year order, with 1 added to the month and year and 1 subtracted from the day. Thus, if the date were June 24, 1971 the correct signon sequence would be 072372. The signon procedure is illustrated below.

MONITOR transfers control to the modules INDEXING, SEARCH, and THESAURI via the sequences given below. The user entering the command "exit" causes the module in control to pass control back to MONITOR.

```
# $run *time
# 15:19.10
  CLOCK 15:19.10  DATE  06-24-71
# 15:19.11 .016 RC=0
```



```

# 15:15.40
SIGNON CODE?
072372
MODULE DESIRED?
search
DATA BASE?
exit
MODULE DESIRED?
indexing
ACCESS COMMAND?
exit
MODULE DESIRED?
thesn
COMMAND?
exit
MODULE DESIRED?
theso
COMMAND?
exit
MODULE DESIRED?
stop
# 15:17.21 .921 RC=43096

```

5.6.3 THESAURI

In all of the commands recognizable by the module THESAURI only the first four characters must be correctly specified. This does not stop the user from entering the complete associated word if he so desires. The following are sample commands recognizable by the module THESAURI. The response from the module THESAURI follows the user entered commands.

(1) Enter the term HAIL into the thesaurus along with the following terms and associated relationships: ICE STONES (used for - 3) and GRAUPEL (related term - 6). Striking the carriage return key or entering any single character causes control to be returned to the command recognition routine.

```

COMMAND?
term
hail

```



```
ice stones3
graupel6
c
COMMAND?
```

(2) Print the complete thesaurus (terms and relationships, and reference numbers and classification codes if any).

```
COMMAND?
display
complete
GRAUPEL 0002 551.578.7:
    RT HAIL
HAIL 0001 551.578.7:
    UF ICE STONES
    RT GRAUPEL
ICE STONES
    USE HAIL
COMMAND?
```

(3) Print alphabetical listing of all thesaurus entries.

```
COMMAND?
display
alphabetic
GRAUPEL
HAIL
ICE STONES
COMMAND?
```

(4) Print one ("pone") thesaurus entry and relationships and thesaurus reference number if any.

```
COMMAND?
display
pone
hail
HAIL 0001
    UF ICE STONES
    RT GRAUPEL
COMMAND?
```

(5) Determine how many entries are present in the thesaurus, and how much space is available between the thesaurus tables.

```
COMMAND?
space
00003 TERMS
```


32682 BYTES
COMMAND?

(6) Change the spelling of the entry HAIL to HALE and determine if the spelling change was made correctly. HAIL was initially entered into the thesaurus under point (1) above.

COMMAND?
spelling
hail
hale
COMMAND?
display
pone
hail
NOT IN FILE
COMMAND?
display
pone
hale
HALE 0001
UF ICE STONES
RT GRAUPEL
COMMAND?

(7) Delete the term HALE and all relationships. The "d" following the term indicates that the term and all relationships are to be deleted.

COMMAND?
delete
haled
COMMAND?

(8) Delete the "used for" relationship ICE STONES from the term HAIL. A blank must be entered by the user following the term HAIL before carriage return to indicate that relationships for the term are to be deleted.

COMMAND?
delete
hail
ice stones3
COMMAND?

(9) Move the internal program table TERMSTOR up ("u") 10 positions.

```
COMMAND?
move
u10
COMMAND?
```

5.6.4 INDEXING

The module INDEXING handles record manipulation in the five data bases in the programming system. INDEXING allows for:

(1) retrieving records for display purposes; (2) updating records; and (3) entering new records. In the examples given the operations are conducted on dummy data bases.

(1) Enter a record ("n") into the fifth record ("0005") in the data base ("keyd") which consists of document numbers and keywords used in indexing the documents.

```
ACCESS COMMAND?
keyd0005n
INDEXING ENTRY?
2222*hail *graupe*precip*
ACCESS COMMAND?
```

(2) Retrieve ("r") the record mentioned in (1).

```
ACCESS COMMAND?
keyd0005r
2222*HAIL *GRAUPE*PRECIP*
ACCESS COMMAND?
```

(3) Update ("u") the record mentioned in (1) to contain another keyword which indexes the document.

```
ACCESS COMMAND?
keyd0005u
2222*HAIL *GRAUPE*PRECIP*
INDEXING ENTRY?
2222*hail *graupe*precip*freezi*
ACCESS COMMAND?
```


(4) Retrieve ("r") the updated record.

```
ACCESS COMMAND?
keyd0005r
2222*HAIL *GRAUPE*PRECIP*FREEZI*
ACCESS COMMAND?
```

(5) Attempt to retrieve ("r") a record which is not present in the data base being considered.

```
ACCESS COMMAND?
udcn0004r
NO ENTRY IN FILE
ACCESS COMMAND?
```

(6) Enter a record ("n") into the second record ("0002") in the data base ("udcn") which consists of U.D.C. classification codes corresponding to a thesaurus entry. The record number in the U.D.C. classification code file is the thesaurus reference number assigned to the thesaurus entry by THESAURI.

```
ACCESS COMMAND?
udcn0002n
INDEXING ENTRY?
636/639(28)(210):
ACCESS COMMAND?
```

(7) Retrieve ("r") the record mentioned in (6).

```
ACCESS COMMAND?
udcn0002r
636/639(28)(210):
ACCESS COMMAND?
```

(8) Enter a record ("n") into the fourth record ("0004") in the data base ("udcd") which consists of document numbers and classification codes used in indexing the documents.

```
ACCESS COMMAND?
udcd0004n
INDEXING ENTRY?
2222*636/639(28)(210):
ACCESS COMMAND?
```


(9) Retrieve ("r") the record mentioned in (8).

```
ACCESS COMMAND?
udcd0004r
2222*636/639(28)(210):
ACCESS COMMAND?
```

(10) Enter a record ("n") into the third record ("0003") in the data base ("refn") which consists of document numbers and thesaurus reference numbers which correspond to indexing entries.

```
ACCESS COMMAND?
refn0003n
INDEXING ENTRY?
2222*0021*0025*0037*0082*
ACCESS COMMAND?
```

(11) Retrieve ("r") the record mentioned in (10).

```
ACCESS COMMAND?
refn0003r
2222*0021*0025*0037*0082*
ACCESS COMMAND?
```

When making large additions to any of the files the following MTS commands may be used:

```
$copy *source* to filename("starting record number")
  }
  } images or records to be added to filename
  }
$endfile
```

5.6.5 SEARCH

The module SEARCH handles the searching of data bases in the programming system utilizing the techniques discussed in Section 5.3.5.

In the examples given the document numbers (if any) which are "hits" for the questions are returned. The searches were

conducted on dummy data bases.

(1) Search conducted on the data base which consists of document numbers and keywords used in indexing the documents.

```
DATA BASE?
keyword
ENTER QUESTIONS
profile*01*
and*hail  *01*
or *ice  *01*
end
SORT=CH;D;1;4 INPUT=-SORTINP(1,006) F;256;256 OUTPUT=-SORTOUT .....
STATISTICS:      6/  0
    PROFILE*01*
    AND*HAIL  *01*
    OR *ICE    *01*
    ** DOCUMENT NUMBER 2251 **
    ** DOCUMENT NUMBER 2250 **
    ** DOCUMENT NUMBER 2253 **
DATA BASE?
```

(2) Searches conducted on the data base which consists of document numbers and thesaurus reference numbers.

```
DATA BASE?
refnumb
ENTER QUESTIONS
profile*01*
and*0010*01*
profile*01*
and*01*
end
SORT=CH;D;1;4 INPUT=-SORTINP(1,007) F;256;256 OUTPUT=-SORTOUT .....
STATISTICS:      7/  0
    PROFILE*01*
    AND*0010*01*
    ** DOCUMENT NUMBER 2222 **
    ** DOCUMENT NUMBER 2229 **
    ** DOCUMENT NUMBER 2228 **
    PROFILE*01*
    AND*01*
** ERROR -- SEE COM.....
```

```
DATA BASE?
refnumb
ENTER QUESTIONS
profile*01*
and*0010*01*
not*0001*01*
end
```



```

SORT=CH;D;1;4 INPUT=-SORTINP(1,004) F;256;256 OUTPUT=-SORTOUT .....
STATISTICS:      4/    0
    PROFILE*01*
    AND*0010*01*
    NOT*0001*01*
    ** DOCUMENT NUMBER 2222 **
DATA BASE?

```

(3) Search conducted on the data base which consists of document numbers and classification codes used in indexing the documents. This example illustrates the signon procedure and the means of accessing the module SEARCH.

```

# 13:49.18
SIGNON CODE?
072472
MODULE DESIRED?
search
DATA BASE?
classed
ENTER QUESTIONS
profile*01*
and*636/639*01*  animals
profile*01*
and*312.8*01*  human factor
profile*01*
and*389.151*01*  metric units
end
SORT=CH;D;1;4 INPUT=-SORTINP(1,011);F;256;256 OUTPUT=-SORTOUT ....
STATISTICS:      11/    0
    PROFILE*01*
    AND*636/639*01*  ANIMALS
    ** DOCUMENT NUMBER 2228 **
    ** DOCUMENT NUMBER 2230 **
    ** DOCUMENT NUMBER 2229 **
    PROFILE*01*
    AND*312.8*01*  HUMAN FACTOR
    ** DOCUMENT NUMBER 2224 **
    PROFILE*01*
    AND*389.151*01*  METRIC UNITS
    ** DOCUMENT NUMBER 2222 **
DATA BASE?

```


5.7 Conclusions

Every attempt was made to keep the query language from becoming too involved in the design and coding of the modules in the system. This was done to eliminate, as far as possible, the annoyance caused by having to wait while the slow process of printing user instructions takes place. This characteristic of the system means, however, that the user must be aware of what is expected of him before he begins to use the system. The orientation process would be quite brief. Sessions conducted with students using the programs written by Sohnle and Wong proved that learning takes place quickly.

Precautionary steps guarding against possible destruction of information vital to system operations are necessary in any implementation of a system to which many users have access. In this system the signon code feature almost totally eliminates the possibility of an unauthorized user being allowed to alter information in system files. For further safeguarding a tape backup should be used.

The general nature of the modules in the system allows for reasonably easy modification if expansion is desired. With all I/O operations performed by one module (IOMODULE) major changes are confined to this module in the event that the system is implemented under another time-sharing system.

CHAPTER VI

FUTURE CONSIDERATIONS

6.1 Introduction

The system developed for this thesis, while being operative as it presently exists, can be used as the starting point for further developments in the area of thesaurus utilization. In this chapter expansion possibilities are discussed and specific storage requirements for the operation of THESAURI are given.

6.2 Storage Requirements

The module THESAURI requires some of the information to be "in core" and some of the information to be stored on disk.

For each term entered into the thesaurus via THESAURI the computer memory requirements are as follows: (1) TERMLIST ENTRY -- 8 bytes; (2) TERMSTOR entry -- number of bytes required to store term (call this TL). Thus, in a thesaurus of N terms the amount of computer memory required would be: $N(8 + TL)$ bytes. In the Water Resources Thesaurus [84] in which there are approximately 5200 terms with an average length of 12 characters the amount of memory required is:

$$\begin{aligned} 5200 (8 + 12) &= 5200(20) \\ &= 104000 \text{ bytes} \\ &\approx 100 \text{ K.} \end{aligned}$$

THESAURI uses records up to 256 bytes in length for relationship accounting. If more than 256 bytes are required to account for the relationships for a thesaurus entry a second record up to

256 bytes in length is used. However, usually less than 256 bytes are required to account for relationship information for a thesaurus entry.

For a thesaurus entry for which there are: (1) m different types of relationships (broader term, related term, etc.); (2) n_i , $i = 1, 2, \dots, m$ terms related to the thesaurus entry by the designated relationship; the space required is: (1) information regarding type of relationship -- 8 bytes for each relationship; (2) information regarding each entry related to the main entry by the designated relationship -- 8 bytes; or, in symbols:

$$\begin{aligned}
 & 8m + \sum_{i=1}^m 8n_i \\
 &= 8m + 8 \sum_{i=1}^m n_i \\
 &= 8 \left[m + \sum_{i=1}^m n_i \right] \text{ bytes.}
 \end{aligned}$$

Thus, for a term for which there: (1) exists 3 different types of relationships; (2) are 2 terms related to the main entry by the first relationship, 3 terms related to the main entry by the second relationship, and 4 terms related to the main entry by the third relationship; the disk space required is:

$$\begin{aligned}
 & 8 \left[m + \sum_{i=1}^m n_i \right] \\
 &= 8 [3 + (2 + 3 + 4)] \\
 &= 8 [3 + 9] \\
 &= 96 \text{ bytes.}
 \end{aligned}$$

6.3 Specification of a Data Base Format

If the system were employed in an information center it is desirable that the system be used in searching data bases from other centers or institutions. For this reason it is desirable to have a record format specification for information which is to be operated on by the system. A typical format specification might allow for information such as: (1) author(s); (2) title; (3) keywords representing document content; (4) abstract; (5) classification codes representing document content; (6) date of publication; (7) accession number.

When incorporating a new data base into this system the records in the data base would be converted into the specified format. This, of course, would be done through use of an appropriate computer program. Computer programs existing as part of the present system could then be used to convert this format into the subsets suitable for handling by the system. The only additional programming required would be to convert the data base into the format required by the system.

Thus, if a person wished to use with the system a tape containing information not formatted in the specified manner, the essential steps to be taken would be: (1) write a computer program to convert the incoming tape into the specified format; (2) after the tape has been converted into the specified format use programs that exist as part of the system to create the required data bases; (3) use the computer programs that now exist to operate on these data bases.

This type of implementation would speed operations and information would be made available to users with shorter time delays.

6.4 Modification of Method for Obtaining Thesaurus Reference Number

A place where immediate modification could be made to systems design would be a modification of the method for obtaining thesaurus reference number.

At present the thesaurus reference numbers are assigned sequentially, from zero, as new indexing and searching entries are entered into the thesaurus. The assigned numbers are stored as a two byte portion of a TERMLIST entry. These numbers then serve as record numbers in an MTS line file where classification codes for the thesaurus entry in question could be stored and also as entries in the data base consisting of document information indexed by thesaurus reference numbers which correspond to keywords.

By employing a hashing technique [4] on the thesaurus entry being considered the necessity of storing the thesaurus reference number as part of a TERMLIST entry is eliminated. When printing a thesaurus entry where the thesaurus reference number is required a routine which computes the hash code could be activated to compute the code for the thesaurus entry in question. The method of computation of the code would be such that the computed codes could correspond to record numbers in an MTS file.

One problem that arises when utilizing a scheme of this type is the problem of collisions, or more explicitly, having more than one key map into the same hash address. In reference to this system

a method of distinguishing between duplicate references to classification codes for any thesaurus references would have to be resolved. An immediate solution to the problem would be to reformat the records in the MTS file containing classification codes corresponding to thesaurus entries so they would appear as illustrated below.

| | | |
|---------------------|--------------------|------------------------------------|
| 2 byte indicator | thesaurus entry | corresponding classification codes |
|---------------------|--------------------|------------------------------------|

The "2 byte indicator" contains -1 if only one thesaurus entry maps into this record number or the record number of another record in the file where classification codes for a thesaurus entry mapping into this same record address are located. The "thesaurus entry" portion of the record would contain the thesaurus entry with which the classification codes are associated. A comparison against the specified entry would have to be made in order to determine correspondence. Records would have to be read until the specified entry and the term in the entry portion of the record correspond. One problem specific to this suggestion would be deciding how many bytes should be allowed for the "thesaurus entry" portion of the disk record. As in any implementation of hash coding one of the main problems that arises is deciding how the records are obtained for use after collisions occur.

6.5 Modification of Method for Obtaining Record Numbers for Storage of POINTABL Information

Hashing could also be used to obtain the record numbers which indicate the position of relationship information for thesaurus

entries. However, the same problems would arise as are mentioned in the discussion of the use of hash coding to determine thesaurus reference numbers.

The use of hashing to determine "record numbers" and "reference numbers" would eliminate four of the bytes in a TERMLIST entry. The core space required to store a thesaurus of N terms with an average length of TL bytes would be: $N(4 + TL)$ bytes. The Water Resources Thesaurus would require:

$$\begin{aligned} 5200 (4 + 12) &= 5200(16) \\ &= 83200 \text{ bytes} \end{aligned}$$

or a savings of: $104000 - 83200 = 20800$ bytes.

Program complexity, however, would definitely be increased.

6.6 Automatic Modification of Search Strategy

The system, as it presently operates, is wholly dependent on the user for specification of search terms when searches are to be conducted on data bases. In any implementation of a thesaurus based system a desired feature would be the automatic expansion of search strategy to include other thesaural relationships depending on user specified options. The possible options might include:

(1) expansion to include narrower terms; (2) expansion to include related terms; (3) expansion to include broader terms. Two systems employing the automatic expansion of search strategy were mentioned in Section 2.6.4.

The inclusion of this feature in the system would require modifications to the modules THESAURI and SEARCH. THESAURI would

have to be modified to allow its being used both in the manner it is presently used and also to allow its being called from the module SEARCH. The thesaurus entry for which expansion of search strategy is to be made would have to be accessed by THESAURI. However, rather than printing the entry and its relationships for the purpose of viewing by the user the information would be stored in a designated area in the module SEARCH. The relationship entries corresponding to the expansion strategy desired would have to be added to the tables set up by SEARCH. SEARCH would be responsible for applying these entries to the correct question using the weight specified on the term originally used in the question, with OR logic.

With this capability the system would be very powerful.

6.7 Application of Coding

In use with a large data base, to speed operation of the module SEARCH, coding schemes could be applied. At the University of Alberta H.S. Heaps and L.H. Thiel [32] have done work in this area.

Rather than searching on the data base entries themselves question entries would be coded before being entered into tables for subsequent testing of correspondence with coded data base entries. If automatic expansion of search strategy were employed the coded thesaurus entries might be included as part of the thesaurus entry so conversion from term to code could be eliminated in any expansion of search strategy.

CHAPTER VII

CONCLUSIONS

The main aim of this research was to develop efficient computer programs to allow for thesaural manipulations in an on-line environment. The establishment of nationwide information networks makes necessary effective retrieval and dictates use of thesauri. The importance of vocabulary control through thesauri and similar word authority lists will increase and future extension will make inevitable the need for sophisticated computer programs for thesauri control.

Further, the linkage of a thesaurus and a classification scheme provides a research tool through which thesaurus making and classification building can be explored, and, when joined with other information storage and retrieval capabilities, furnishes a means whereby the suitability of various indexing and classification techniques for specific areas of information can be ascertained.

The importance of on-line access for indexing and classification, although not discussed in detail, is regarded as self evident. It should be noted, however, that this assumption can only be justified when the efficiency and utility of such on-line manipulation is ascertained after a period of controlled testing. Such testing was beyond the scope of this thesis.

BIBLIOGRAPHY

1. Agricultural/Biological Vocabulary, Washington, D.C., United States Department of Agriculture, 1967.
2. Aitchison, J., "The Thesaurofacet: A Multipurpose Retrieval Language Tool", Journal of Documentation, Vol. 26, No. 3, pp. 187-203, September 1970.
3. Aitchison, J., A. Gomersall, and R. Ireland, Thesaurofacet A Thesaurus and Faceted Classification for Engineering and Related Subjects, Whetstone, England, English Electric Company Ltd., 1969.
4. Alber, F.M., A Survey of Hash Coding and Associated Problems, University of Alberta, Edmonton, Department of Computing Science, Unpublished Term Paper, 1971.
5. Atherton, P. and K.B. Miller, "LC/MARC on MOLDS; An Experiment in Computer-Based, Interactive Bibliographic Storage, Search, Retrieval, and Processing", Journal of Library Automation, Vol. 3, No. 2, pp. 142-165, June 1970.
6. Backer, S. and E.I. Valko, Thesaurus of Textile Terms, Cambridge, Massachusetts, The M.I.T. Press, 1969.
7. Barhydt, G.C., C.T. Schmidt, and K.T. Chang, Information Retrieval Thesaurus of Education Terms, Cleveland, Ohio, Case Western Reserve Press, 1968.
8. Bernier, C.L. and K.F. Heumann, "Correlative Indexes III, Semantic Relations Among Semantemes -- The Technical Thesaurus", American Documentation, Vol. 8, No. 3, pp. 211-220, July 1957.
9. Blagden, J.F., "Thesaurus Compilation Methods: A Literature Review", ASLIB Proceedings, Vol. 20, No. 8, pp. 345-359, August 1968.
10. Bureau of SHIPS Thesaurus of Descriptive Terms and Code Book, Washington, D.C., Bureau of SHIPS, Navy Department, 1965.
11. Cain, A.M., "Thesaural Problems in an On-Line System", Medical Library Association Bulletin, Vol. 57, No. 3, pp. 250-259, July 1969.
12. Caponio, J.F. and T.L. Gillum, "Practical Aspects Concerning the Development and Use of ASTIA's Thesaurus in Information Retrieval", Journal of Chemical Documentation, Vol. 4, pp. 5-8, 1964.

13. Clampett, H.A. Jr., "Randomized Binary Searching with Tree Structures", Communications of the ACM, Vol. 7, No. 3, pp. 163-165, March 1964.
14. Clough, C.R. and K.M. Bramwell, "A Single Computer Based System for Both Current Awareness and Retrospective Search: Operating Experience with ASSASSIN", Paper Presented at Cranfield Mechanized Information Conference, July 1971.
15. Colgan, F.J.J., Report of the Literature Treating of the Major Fields of Development in Thesaurus Work During the Past Two Years, University of Alberta, Edmonton, Department of Computing Science, Unpublished Term Paper, 1970.
16. Costello, J.C. Jr., "An Introduction to Deep (Co-ordinate) Indexes", Journal of Chemical Documentation, Vol. 3, pp. 163-167, 1963.
17. Costello, J.C. Jr., Systems for the Intellectual Organization of Information Volume VII Coordinate Indexing, New Brunswick, New Jersey, The Rutgers University Press, 1966.
18. Davis, C.H., "Integrating Vocabularies with a Classification Scheme", American Documentation, Vol. 19, No. 1, p. 101, January 1968.
19. Dennis, S.F., "The Construction of a Thesaurus Automatically from a Sample of Text", Statistical Association Methods for Mechanized Documentation, Proceedings of the Symposium held in March 1964 in Washington, D.C., Washington, D.C., pp. 61-148, 1964.
20. Department of Defense Manual for Building a Technical Thesaurus, Project LEX, Office of Naval Research, 1966, Clearinghouse Number AD 633 279.
21. Dodd, G.G., "Elements of Data Management Systems", Computing Surveys, Vol. 1, No. 2, pp. 117-133, June 1969.
22. Dowell, N.G. and J.W. Marshall, "Experience with Computer Produced Indexes", ASLIB Proceedings, Vol. 14, No. 10, pp. 323-332, October 1962.
23. Eichhorn, M.M. and R.D. Reinecke, "Development and Implementation of a Thesaurus for the Visual Sciences", Journal of Chemical Documentation, Vol. 9, No. 2, pp. 114-118, May 1969.
24. Eller, J.L. and R.L. Panek, "Thesaurus Development for a Decentralized Information Network", American Documentation, Vol. 19, No. 3, pp. 213-220, July 1968.

25. Euratom Thesaurus, Brussels, Belgium, European Atomic Energy Community, 1966.
26. Freeman, R.R., "Computers and Classification Schemes", Journal of Documentation, Vol. 20, No. 3, pp. 137-145, 1964.
27. Freeman, R.R., "The Management of a Classification Scheme", Journal of Documentation, Vol. 23, No. 4, pp. 304-320, December 1967.
28. Gaster, K., "Thesaurus Construction and Use: A Selective Bibliography Based on Material in the ASLIB Library in July 1967", ASLIB Proceedings, Vol. 19, No. 9, pp. 310-317, September 1967.
29. Gillum, T.L., "Compiling a Technical Thesaurus", Journal of Chemical Documentation, Vol. 4, pp. 29-32, 1964.
30. Grosch, A.N., "Thesaurus Construction", Special Libraries, Vol. 60, No. 2, pp. 87-92, February 1969.
31. Hargrave, C.W. and E. Wall, "Retrieval Improvement Effected by Use of a Thesaurus", Proceedings of American Society for Information Science, Vol. 7, pp. 291-294, 1970.
32. Heaps, H.S. and L.H. Thiel, "Optimum Procedures for Economic Information Retrieval", Information Storage and Retrieval, Vol. 6, No. 2, pp. 137-154, June 1970, (Continued Vol. 7, No. 4, December 1971).
33. Herner, S., F.W. Lancaster, and W.F. Johanningsmeier, "A Case Study in the Application of Cranfield System Evaluation Techniques", Journal of Chemical Documentation, Vol. 5, pp. 92-95, 1965.
34. Hersey, D.F. and W. Hammond, "Computer Usage in the Development of a Water Resources Thesaurus", American Documentation, Vol. 18, No. 4, pp. 209-215, October 1967.
35. Holm, B.E. and L.E. Rasmussen, "Development of a Technical Thesaurus", American Documentation, Vol. 12, No. 3, pp. 184-190, July 1961.
36. Horvath, P.J., A.Y. Chamis, R.F. Carroll, and J. Dlugos, "The B.F. Goodrich Information Retrieval System and Automatic Distribution Using Computer Compiled Thesaurus and Dual Dictionary", Journal of Chemical Documentation, Vol. 7, No. 3, pp. 124-130, August 1967.
37. Information Retrieval Subject Authority List, American Petroleum Institute, 1968.

38. Joyce, T. and R.M. Needham, "The Thesaurus Approach to Information Retrieval", American Documentation, Vol. 9, No. 3, pp. 192-197, 1958.
39. Lancaster, F.W., "Evaluating the Small Information Retrieval System", Journal of Chemical Documentation, Vol. 6, pp. 158-160, 1966.
40. Lancaster, F.W. and W.F. Johanningsmeier, Project SHARP Information Storage and Retrieval System: Evaluation of Indexing Procedures and Retrieval Effectiveness, Washington, D.C., Department of Navy, Bureau of SHIPS, 1964.
41. Lancaster, F.W. and J. Mills, "Testing Indexes and Index Language Devices: The ASLIB Cranfield Project", American Documentation, Vol. 15, No. 1, pp. 4-13, January 1964.
42. London, G., A Classed Thesaurus as an Intermediary Between Textual and Searching Languages, New Brunswick, New Jersey, The State University Graduate School of Library Service Bureau of Information Sciences Research, 1966.
43. Maixner, V., "Towards Compensation Laws in Constructing Thesauri", Information Storage and Retrieval, Vol. 6, No. 5, pp. 383-386, December 1970.
44. Mandersloot, W.G.B., E.M.B. Douglas, and N. Spicer, "Thesaurus Control - The Selection, Grouping and Cross-Referencing of Terms for Inclusion in a Co-ordinate Index Word List", ASIS Journal, Vol. 21, No. 1, pp. 49-57, January-February 1970.
45. Martinez, S.J., L.P. Brown, D.P. Helander, and H.O. McLeod, "Computer Processing of Thesaurus Data", Proceedings of American Society for Information Science, Vol. 6, pp. 269-275, 1969.
46. Martinez, S.J. and D.P. Helander, "The Development and Maintenance of a Specialized, Controlled Vocabulary Thesaurus", Proceedings of American Society for Information Science, Vol. 5, pp. 277-283, 1968.
47. Matthews, F.W. and L. Thomson, "Weighted Term Search: A Computer Program for an Inverted Co-ordinate Index on Magnetic Tape", Journal of Chemical Documentation, Vol. 7, No. 1, pp. 49-56, February 1967.
48. Mercier, M.A., "Study of U.D.C. and other Indexing Languages through Computer Manipulation of Machine Readable Data Bases", University of Alberta, Edmonton, Master's Thesis, 1971.

49. Neville, H.H., "Feasibility Study of a Scheme for Reconciling Thesauri Covering a Common Subject", Journal of Documentation, Vol. 26, No. 4, pp. 314-336, December 1970.
50. Oller, R.G., Human Factors Data Thesaurus (An Application to Task Data), Wright-Patterson Air Force Base, Ohio, System Development Corporation, 1968, Clearinghouse Number AD 670 578.
51. Papier, L.S. and E.H. Cortelyou, "Use of a Technical Word Association Test in the Preparation of a Thesaurus", Journal of Documentation, Vol. 18, No. 4, pp. 183-187, December 1962.
52. Patt, Y.N., "Variable Length Tree Structures Having Minimum Average Search Time", Communications of the ACM, Vol. 12, No. 2, pp. 72-76, February 1969.
53. Rainey, L., "Experience with the New TEST Thesaurus and the New NASA Thesaurus", Special Libraries, Vol. 61, No. 1, pp. 26-32, January 1970.
54. Robinson, F., "A Computer Based Retrieval System Using a Reactive Thesaurus", Information Storage and Retrieval, Vol. 6, No. 2, pp. 171-189, June 1970.
55. Roget's Pocket Thesaurus, Richmond Hill, Ontario, Simon and Schuster of Canada, 1970, 94th Printing.
56. Rolling, L.N., "Compilation of Thesauri for Use in Computer Systems", Information Storage and Retrieval, Vol. 6, No. 4, pp. 341-351, October 1970.
57. Rostron, R.M., "The Construction of a Thesaurus", ASLIB Proceedings, Vol. 20, No. 3, pp. 181-187, March 1968.
58. Salton, G., Automatic Information Organization and Retrieval, New York, McGraw-Hill Book Company, 1968.
59. Sasamori, K., "Software Design for Vocabulary Control (DOCTOR) System", Proceedings of American Society for Information Science, Vol. 7, pp. 195-197, 1970.
60. Schirmer, R.F., "Thesaurus Analysis for Updating", Journal of Chemical Documentation, Vol. 7, No. 2, pp. 94-98, May 1967.
61. Schultz, C.K., Thesaurus of Information Science Terminology, Washington, D.C., Communication Service Corporation, 1968.
62. Schultz, C.K., P.D. Schwartz, and L. Steinberg, "A Comparison of Dictionary Use Within Two Information Retrieval Systems", American Documentation, Vol. 12, No. 4, pp. 247-253, October 1961.

63. Secretant, M. (editor), Bulletin De L'Association Internationale Des Documentalistes, Paris, France, Gauthier-Villars, Vol. V, No. 4, 1966.
64. Shepherd, C.A., "The Computer-Stored Thesaurus and its Use in Concept Processing", Proceedings AFIPS 1963 Fall Joint Computer Conference, pp. 389-395, 1963.
65. Smilari, S., "Structure, Preparation, and Computer Input of a Biomedical and Medical Thesaurus for Syntex Research", Proceedings of American Society for Information Science, Vol. 5, pp. 289-292, 1968.
66. Sohnle, R.C., University of Alberta, Edmonton, Department of Computing Science, Personal Communication, 1971.
67. Sommar, H.G. and D.E. Dennis, "A New Method of Weighted Term Searching with a Highly Structured Thesaurus", Proceedings of American Society for Information Science, Vol. 6, pp. 193-198, October 1969.
68. Sparck Jones, K., "Automatic Thesaurus Construction and the Relation of a Thesaurus to Indexing Terms", ASLIB Proceedings, Vol. 22, No. 5, pp. 226-228, May 1970.
69. Sparck Jones, K., "Some Thoughts on Classification for Retrieval", Journal of Documentation, Vol. 26, No. 2, pp. 89-101, June 1970.
70. Stone, D.C. and M. Rubinoff, "Statistical Generation of a Technical Vocabulary", American Documentation, Vol. 19, No. 4, pp. 411-412, October 1968.
71. Surace, C.J., The Displays of a Thesaurus, Santa Monica, California, The Rand Corporation, 1970, Clearinghouse Number AD 703 593.
72. Sussenguth, E.H. Jr., "Use of Tree Structures for Processing Files", Communications of the ACM, Vol. 6, No. 5, pp. 272-279, May 1963.
73. Swets, J.A., "Effectiveness of Information Retrieval Methods", American Documentation, Vol. 20, No. 1, pp. 72-89, January 1969.
74. Tancredi, S.A. and O.D. Nichols, "Air Pollution Technical Information Processing - The Microthesaurus Approach", American Documentation, Vol. 19, No. 1, pp. 66-70, January 1968.
75. Taube, M., "Extensive Relations as the Necessary Condition for the Significance of 'Thesauri' for Mechanized Indexing", Journal of Chemical Documentation, Vol. 3, No. 3, pp. 177-180, July 1963.

76. Taube, M., "A Note on the Pseudo-Mathematics of Relevance", American Documentation, Vol. 16, No. 2, pp. 69-72, April 1965.
77. Thesaurus of ASTIA Descriptors, Arlington, Virginia, Armed Services Technical Information Agency, 1962.
78. Thesaurus of Engineering and Scientific Terms, New York, Engineers Joint Council, 1969.
79. Thesaurus of ERIC Descriptors, Washington, D.C., U.S. Government Printing Office, 1968.
80. Thesaurus of Pulp and Paper Terms, Pointe Claire, Quebec, Pulp and Paper Research Institute of Canada, 1965.
81. Thiel, L.H. and H.S. Heaps, Computer Search of Chemical Titles at the University of Alberta, University of Alberta, Edmonton, Department of Computing Science, 1968, Publication No. 15.
82. Vickery, B.C., "Thesaurus -- A New Word in Documentation", Journal of Documentation, Vol. 16, No. 4, pp. 181-189, December 1966.
83. Wall, E., "Vocabulary Building and Control Techniques", American Documentation, Vol. 20, No. 2, pp. 161-164, April 1969.
84. Water Resources Thesaurus, Washington, D.C., United States Department of the Interior Office of Water Resources Research, 1966.
85. Watts, H.N., A Thesaurus for Health, Physical Education and Recreation, University of Alberta, Edmonton, 1969.
86. Wolff-Terroine, M., N. Simon, and D. Rimbert, "Use of a Computer for Compiling and Holding a Medical Thesaurus", Methods of Information in Medicine, Vol. 8, No. 1, pp. 34-40, January 1969.
87. Wong, A.L.S. and D.M. Heaps, 'THESAURI' An On-Line Program for Constructing a Thesaurus, University of Alberta, Edmonton, Department of Computing Science, 1969, Publication No. 20.

APPENDIX

The special symbols used in defining the command language are given in Table A-1.

| <u>Symbol</u> | <u>Meaning</u> |
|---------------|------------------|
| ::= | is defined to be |
| | or |
| < > | a name |
| [] | optional |
| { } | choose one of |

Table A-1: Symbols Used in Command Language

The symbols not enclosed in angular brackets are literals and represent themselves.

In the specifications certain entities have not been defined by basic symbols. In these instances an explanation is given in the semantics section. The number to the left of the expression is the number of the explanation.

The specifications have been grouped so that commands for a particular module in the programming system appear together.

Basic Symbols

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<letter> ::= <any non-digit character>

<character sequence> ::= <letter> | <digit> | <character sequence>

<letter> | <character sequence> <digit>

<character> ::= <letter> | <digit>

Module MONITOR

```

1  <signon code>::= <nr> <signon sequence> <run code>

    <run code>::= <batch mode designation> | <terminal mode designation>

    <batch mode designation>::= B

    <terminal mode designation>::= [T]

    <module designation>::= <nr> { <indexing designation> | <search
                                designation> | <thesaurus designation> <thesaurus
                                code> | <system stop designation> }

    <indexing designation>::= INDE

    <search designation>::= SEAR

    <thesaurus designation>::= THES

    <thesaurus code>::= <new code> | <old code>

    <new code>::= N

    <old code>::= O

    <system stop designation>::= STOP

```

Module THESAURI

```

    <thesaurus command>::= <nr> { <display designation> | <module exit
                                designation> | <enter term designation> | <delete
                                designation> | <change spelling designation> |
                                <display space designation> | <move TERMSTOR
                                designation> | <initialize thesaurus designation> }

    <display designation>::= DISP <display sequence>

    <module exit designation>::= EXIT

    <enter term designation>::= TERM <enter term sequence> <nr>

    <delete designation>::= DELE <delete sequence>

```



```

<change spelling designation>::= SPEL <change spelling sequence>
<display space designation>::= SPAC
<move TERMSTOR designation>::= MOVE <move sequence>
<initialize thesaurus designation>::= INIT
<display sequence>::= <nr> { <complete list designation> |
    <alphabetic list designation> | <display one
    entry designation> }
<complete list designation>::= COMP
<alphabetic list designation>::= ALPH
<display one entry designation>::= PONE <nr> <thesaurus entry>
<thesaurus entry>::= <character sequence>
<enter term sequence>::= <nr> <thesaurus entry> | <enter term
    sequence> <nr> <thesaurus relationship entry>
<thesaurus relationship entry>::= <thesaurus entry> <relationship
    code>
<delete sequence>::= <nr> { <thesaurus entry> <thesaurus delete
    code> | <thesaurus entry> <nr> <thesaurus
    relationship entry> }
<thesaurus delete code>::= D
<change spelling sequence>::= <nr> <thesaurus entry> <nr>
    <thesaurus entry>
<move sequence>::= <nr> { <upward indicator> | <downward
    indicator> } <digit> <digit>
<upward indicator>::= U
<downward indicator>::= D

```


Module INDEXING

<access command>::= <nr> <file designation> <record number>

 <action code>

<file designation>::= <U.D.C. classification codes designation> |
 <keyworded documents designation> | <U.D.C. classified
 documents designation> | <thesaurus reference numbers
 documents designation> | <bibliographic data base
 designation> | <module exit designation>

<U.D.C. classification codes designation>::= UDCN

<U.D.C. classified documents designation>::= UDCD

<keyworded documents designation>::= KEYD

<thesaurus reference numbers documents designation>::= REFN

<bibliographic data base designation>::= DATA

<record number>::= <digit> <digit> <digit> <digit>

<action code>::= <retrieve record designation> | <update record
 designation> | <new record record designation>

<retrieve record designation>::= R

<update record designation>::= U

<new record designation>::= N

³ <data base entry>::= <U.D.C. code record> | <keyworded document
 record> | <U.D.C. classified document record> |
 <reference number document record> | <bibliographic
 data base record>

Module SEARCH

```

4 <data base designation>::= <nr> { <keyworded data base
    designation> | <classified data base designation> |
    <reference numbers data base designation> |
    <module exit designation> }

<keyworded data base designation>::= KEYWORD

<classified data base designation>::= CLASSED

<reference numbers data base designation>::= REFNUMB

<profile group>::= <profile> <profile end designation> |
    <profile> <profile group>

<profile end designation>::= <nr> END

<profile>::= <profile designation> <question group>

<profile designation>::= <nr> PROFILE <delimiter> <question
    threshold> <delimiter>

<question threshold>::= <digit> <digit>

<question group>::= <question entry> | <question group>
    <question entry>

<question entry>::= <nr> { <comment entry> | <logic entry> }

<comment entry>::= <comment logic> <comment>

<logic entry>::= { <and logic> | <or logic> | <not logic> }
    <delimiter> <question term> <delimiter> <weight>
    <delimiter>

<and logic>::= AND

<or logic>::= OR

<not logic>::= NOT

```


<comment logic>::= ٠٠٠٠

<weight>::= <digit> <digit>

<delimiter>::= *

<comment>::= <character sequence>

5 <question term>::= <keyword> | <U.D.C. code> | <reference number>

Record Formats: Applicable to the Modules INDEXING and SEARCH

<document accession number>::= <digit> <digit> <digit> <digit>

<keyword>::= <character> <character> <character> <character>
<character> <character>

<reference number>::= <digit> <digit> <digit> <digit>

6 <U.D.C. code record>::= <U.D.C. sequence>

<keyworded document record>::= <nr> <document accession number>
<delimiter> <keyword sequence>

<keyword sequence>::= <keyword> <delimiter> | <keyword sequence>
<keyword> <delimiter>

<reference number document record>::= <nr> <document accession number> <delimiter> <reference number sequence>

<reference number sequence>::= <reference number> <delimiter> |
<reference number sequence> <reference number>
<delimiter>

7 <U.D.C. classified document record>::= <nr> <document accession number> <delimiter> <U.D.C. sequence>

<bibliographic data base record>::= <"free format at present">

Semantics

- ¹ The <signon sequence> is a 6 digit number dependent on the date obtained from the system. In the specifications <nr> means that the program expects a new record. In a batch run this implies that a new card is expected. When utilizing a terminal it means that the carriage must be repositioned to the left margin. This may be done automatically by the program or manually by the user.
- ² The <relationship code> is simply one digit which indicates the relationship of the relationship entry to the specified main entry. The code meanings are dependent on print table entries. This was explained previously in Section 4.4.9.
- ³ By specifying the <file designation> the module INDEXING knows which data base record format is expected. Edit checking, if any, is done according to the specified formats.
- ⁴ The <data base designation> determines the characteristics of the <question term> that the module SEARCH looks for.
- ^{5 6 7} <U.D.C. code> and <U.D.C. sequence> are simply Universal Decimal Classification Codes. Obviously, in the former instance one code is implied while "sequence" implies that U.D.C. codes are separated by colons.

B30010